AD-A039 952 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 13/9
A FEASIBILITY STUDY OF THE HYBRID SIMULATION TECHNIQUE AS APPLI--ETC(U)
APR 77 T L MORAN, J L LEWIS, W H LIVINGSTON

UNCLASSIFIED SPD-767-01 NL

| OF |
AD
A039952

END

DATE
FILMED
6-77

12
B.S.

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

A FEASIBILITY STUDY OF THE HYBRID SIMULATION TECHNIQUE AS APPLIED TO LARGE SCALE CABLE DYNAMICS

by

Thomas L. Moran

Joan L. Lewis

Walter H. Livingston

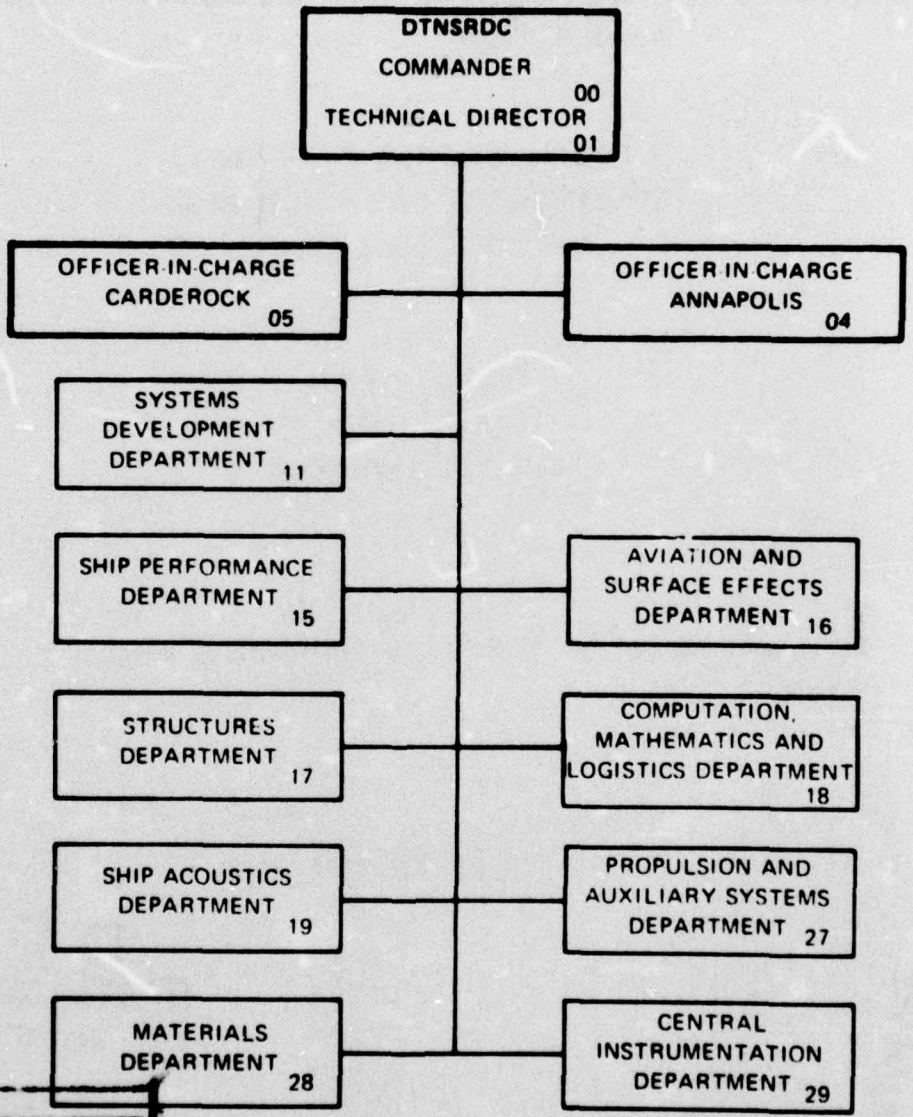
Approved for Public Release: DISTRIBUTION UNLIMITED

SHIP PERFORMANCE DEPARTMENT

April 1977

SPD-767-01

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
(14) 1. REPORT NUMBER SPD-767-01	2. GOVT ACCESSION NO. 3. RECIPIENT'S CATALOG NUMBER
(6) 4. TITLE (and Subtitle) A Feasibility Study of The Hybrid Simulation Technique as Applied to Large Scale Cable Dynamics	5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
(10) 7. AUTHOR(s) Thomas L. Moran, Joan L. Lewis, Walter H. Livingston	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship R&D Center, Code 1576, Bethesda, Maryland 20084	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element, Task Area, Work Unit No. 11576600
11. CONTROLLING OFFICE NAME AND ADDRESS Civil Engineering Laboratory, NCBC, Port Hueneme, CA 93043	(11) 12. REPORT DATE Apr[redacted]977 13. NUMBER OF PAGES 58
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (12) 66 p.	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE n/a

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution Unlimited

(16) N6830[illegible]5 (17) N6830575

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Cable Dynamics
Hybrid
Simulation
Component Time-Sharing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The development of a hybrid computer implementation for the equations of motion for an elastic cable in two dimensions is presented. The discrete element approach is used, in which the partial differential equations that represent the dynamics of the cable system are reduced to a set of 2N ordinary differential equations representing N nodes. A time share procedure is developed in which analog components are shared for different sections of the cable. The basic features of "switching" from one section to the next, and the logic that controls it, are developed using a simple second order — next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

389 694

20. cont.

system of two coupled equations. Then a sample cable problem (six foot buoy release using four nodes) is implemented in full on the hybrid computer, and the results are compared with solutions obtained by an all digital technique.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
ADMINISTRATIVE INFORMATION	iii
INTRODUCTION	1
BRIEF DESCRIPTION OF CABLE PROBLEM	1
BRIEF DESCRIPTION OF SOLUTION PROCEDURES	3
BRIEF DESCRIPTION OF A HYBRID PROGRAM	5
DEVELOPMENT OF TIME-SHARE PROCEDURE	7
HYBRID IMPLEMENTATION OF FOUR-NODE BUOY RELEASE	10
RECOMMENDATIONS	17
DISCUSSION OF VERSATILITY	19
CONCLUSIONS	20
APPENDIX A - Description of Analog Components	A1
APPENDIX B - Detailed Description of Time-Share Procedure	B1
APPENDIX C - Hybrid Program Listings: Analog, Digital, and HOI	C1

LIST OF FIGURES

		Page
Figure 1 -	Solution to the Two Coupled Second Order Dynamic Systems Problem	11
Figure 2 -	Geometry of Four Node Buoy Release Problem	12
Figure 3 -	Equations of Motion for Four Node Buoy Release Problem	13
Figure 4 -	Time History of Cable Node Displacements	16
Figure B1-	Analog Wiring Diagram for Continuous Analog Solution of Two Coupled Second Order Dynamic Systems .	B4
Figure B2-	Analog Wiring Diagram for Component Time Sharing Technique (Employing Track/Store Summers for Storage) for Solution of Two Coupled Second Order Dynamic Systems	B5
Figure B3-	Logic Circuit for Operating Analog Circuits of Figures B1 and B2	B6
Figure B4-	Timing Diagram for Logic of Figure B3	B7
Figure B5-	Analog Wiring Diagram for A/D and D/A Time Sharing Technique for Solution of Two Coupled Second Order Dynamic Systems	B8

ABSTRACT

The development of a hybrid computer implementation for the equations of motion for an elastic cable in two dimensions is presented. The discrete element approach is used, in which the partial differential equations that represent the dynamics of the cable system are reduced to a set of 2N ordinary differential equations representing N nodes. A time-share procedure is developed in which analog components are shared for different sections of the cable. The basic features of "switching" from one section to the next, and the logic that controls it, are developed using a simple second order system of two coupled equations. A sample cable problem (six foot buoy using four nodes) is implemented in full on the hybrid computer, and the results are compared with solutions obtained by an all-digital technique.

ADMINISTRATIVE INFORMATION

The work reported herein was authorized by the Naval Civil Engineering Laboratory. The work request unit was N68305-75-WR-60075. The work was carried out under work unit number 11576600.

INTRODUCTION

Large scale cable dynamics problems, when solved on digital computers, usually run at an inordinately large ratio of required computer seconds to problem seconds. This implies costly solutions. Depending upon the degree of sophisitication with which the cable dynamics are modeled, it would not be uncommon for all-digital solutions to large-scale cable dynamics problems to require several hundred computer seconds per problem second. Such ratios do not encourage design optimization studies via computer simulation.

The purpose of the study reported herein is to explore the feasibility of hybrid computation for the solution of cable dynamics problems, and to show that a hybrid implementation of cable dynamics problems avoids the penalty of long computation times. The following points are addressed in comparing the hybrid approach with the all-digital approach.

a. The accuracies associated with each approach

b. The costs incurred by each approach, where these costs are determined by

   1) The relative sizes of the programming efforts involved

   2) The relative speed of computation as measured by the ratio of computer time to problem time

c. The ease with which parametric studies for design optimization purposes can be handled by each approach

d. The implications of machine dependence incurred by each approach

e. The ease with which, in each approach, essentially the same programming will handle different cable configurations

BRIEF DESCRIPTION OF CABLE PROBLEM

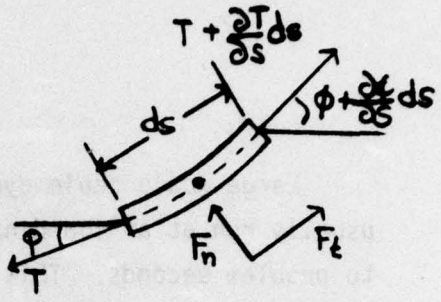
The treatment considered in this report follows that by Ronald Lewis Webster in "An Application of the Finite Element Method to the Determination of Non-Linear Static and Dynamic Responses of Underwater Cable Structures", Cornell University Thesis. It assumes a uniform, continuous, linear, cable material with small strains, where deformation is caused by gravity plus external loads (generally fluid loading). The governing equations, in normal and tangential coordinates, are:

$$\frac{\partial T}{\partial s} = m(\dot{V}_t - V_n\dot{\phi}) + w\sin\phi - F_t$$

$$T\frac{\partial \phi}{\partial s} = m(\dot{V}_n + V_t\dot{\phi}) + w\cos\phi - F_n$$

$$\frac{\partial V_t}{\partial s} = V_n\frac{\partial \phi}{\partial s} + \frac{\dot{T}}{AE}$$

$$\frac{\partial V_n}{\partial s} = V_t\frac{\partial \phi}{\partial s} - \left(1 + \frac{T}{AE}\right)\dot{\phi}$$



where,

m = mass per unit unstretched length

w = weight per unit unstretched length in fluid medium

s = unstretched length

T = tension in the cable

E = Young's modulus

A = cross sectional area of cable

ϕ = angle from the horizontal

$F_{t,n}$ = external tangential, normal load per unit unstretched length

$V_{t,n}$ = tangential, normal cable velocity

To solve the equations of motion of cables, a discrete element approach is usually employed. The effects of mass, internal reactions, and external loads are lumped at a finite number of points (nodes), and the equilibrium and continuity equations are applied to these points, yielding a discrete set of equations. This is, in effect, modeling the continuous system as a set of discrete masses and massless springs. The result is a set of dN coupled differential equations, where N = number of nodes, and d = the dimensionality of the problem (two or three).

In the discrete formulation, the partial differential equations for a two dimensional problem reduce, for node i, to the following set of ordinary differential equations:

$$m_i \ddot{x}_i = -F^N_{i+} \sin\theta_i - F^N_{i-} \sin\theta_i + F^T_{i+} \cos\theta_i + F^T_{i-} \cos\theta_{i-1}$$

$$m_i \ddot{y}_i = +F^N_{i+} \cos\theta_i + F^N_{i-} \cos\theta_{i-1} + F^T_{i+} \sin\theta_i + F^T_{i-} \sin\theta_{i-1} - W_i$$

where,

m_i = mass associated with node i

w_i = weight associated with node i

F^N_{i+} = normal force on node i due to upper half-element

F^N_{i-} = normal force on node i due to lower half-element

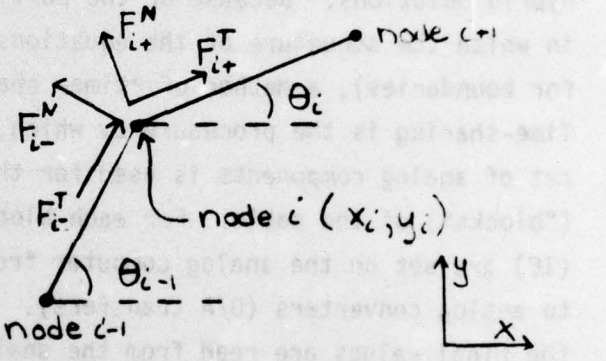
F^T_{i+} = tangential force on node i due to upper half-element

F^T_{i-} = tangential force on node i due to lower half-element

L_i = distance between nodes i+1 and $i_j = \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2\right]^{1/2}$

$\sin\theta_i = (x_{i+1} - x_i)/L_i$

$\cos\theta_i = (y_{i+1} - y_i)/L_i$



The external forces F^N, F^T are generally due to fluid loading, with terms representing added mass, normal and tangential drag, and cable tension.

BRIEF DESCRIPTION OF SOLUTION PROCEDURES

Digital

There are two basic approaches to solving the discrete cable problem on the digital computer. In the explicit scheme, the initial values of each time step are used to compute the values at the end of the step, assuming the inputs to the step are constant within the time step. For an acceptable accuracy, the time step must be very small, leading to very long computation times. In the implicit scheme, values to be solved for in each time step appear in both sides of the equations, which are then solved implicitly by iteration. This procedure again is very time consuming.

Analog and Hybrid

In an attempt to alleviate the problem of long computation times, analog solutions have been investigated. If the number of nodes is small enough, the entire problem can be solved on an analog computer. In general, this is not the case; each node has three degrees of freedom and two integrators per degree of freedom ($\ddot{x}$, $\dot{x}$), that is, six integrators per node. Since a typical analog computer has on the order of 40 integrators, an all-analog problem is limited to less than 10 nodes just on the basis of required number of integrators.

In order to handle larger problems on the analog, one must go to hybrid solutions. Because of the particular nature of the cable problem, in which the structure of the equations for each node is the same (except for boundaries), a method of "time- sharing" analog components is feasible. Time-sharing is the procedure by which, at each time step, the same set of analog components is used for the equations of successive sections ("blocks") of the cable. For each block the appropriate initial conditions (IC) are set on the analog computer from the digital computer by digital to analog converters (D/A transfers). At the end of each time step the final values are read from the analog computer and stored in the digital computer via analog to digital converters (A/D transfers). These final values are used as initial conditions for that block in the next time step. An example of this procedure is given below, where it is assumed that the equations for 5 nodes will fit on the analog, and that there are 10 nodes in the problem: i.e., a block consists of 5 nodes and the problem is defined by 2 blocks.

t_1 $block_1$ = 1) D/A. Set IC's for nodes 1-5, X_i^1, i = 1,5
2) Run analog for Δt
3) A/D. Read final values for nodes 1-5, X_i^2, i = 1,5

t_1 $block_2$ = 1) D/A. Set IC's for nodes 6-10, X_i^1, i = 6,10
2) Run analog for Δt
3) A/D. Read final values for X_i^2, i = 6,10

t_2 block$_1$ = 1) D/A. Set IC's for nodes 1-5, X_i^2, i = 1,5
2) Run analog for Δt
3) A/D. Read final values for nodes 1-5, X_i^3, i=1,5

t_2 block$_2$ = 1) D/A. Set IC's for nodes 6-10, X_i^2, i = 6,10
2) Run analog for Δt
3) A/D. Read final values for nodes 6-10, X_i^3, i = 6,10

et cetera.

BRIEF DESCRIPTION OF A HYBRID PROGRAM

A hybrid program consists of several separate parts:

a. a patch panel on the analog computer,

b. a logic panel on the analog computer,

c. a real time executive program on the digital computer, and

d. an analog setup and checkout program on the digital computer.

The equations to be solved are apportioned between the analog and the digital computers. A separate "HOI" (Hytran Operations Interpreter) program is written that checks on the integrity of analog components, compares computed "static check" values with measured "static check" values, flags wiring errors, and sets all coefficient devices and function generators. The logic panel is designed to send out timing pulses and control signals used for controlling and synchronizing the analog and digital programs. A digital executive program is written that does the following:

a. the necessary setup and initialization of the computer states for real time,

b. A/D and D/A input and output operations,

c. digital computations, and

d. time history storage (on disk or tape), to be processed at the end of the real time run.

In the programs in this report, there is no digital computation or time history storage, although these could be easily added.

The procedure used when actually making a run is:

a. mount the patch panel and logic panel on the analog,

b. run the HOI setup and checkout program,

c. load the real time digital program, and

d. make the run.

A schematic of the logic panel timer, analog state, and digital executive is shown below. The analog state indicates the condition of the analog integrators:

a. IC (initial conditions): integrator outputs are set to initial conditions

b. OP (operate): inputs are being integrated

c. HOLD: integrator outputs are held at their current value.

Logic Timer	Analog State	Executive Program
		Initialize real time system
		Read run input parameters
		Load initial analog values
		Begin real time loop
A	IC	Wait for IC signal (A) from logic panel
		Transfer D/A's
B	OP	Wait for HOLD signal (C)
C	HOLD	Transfer A/D's
		Load next D/A settings
		Test for QUIT signal from logic panel
		Continue loop

In the hybrid time-share mode, successive timer cycles are used successive blocks of the cable, so that with n blocks, there would be n timer cycles for each problem frame time.

There were two major steps to implementing the hybrid time-share approach. The first was to develop a functional time-share procedure. The second was to develop the hybrid program itself. These steps are discussed in the following sections.

DEVELOPMENT OF TIME-SHARE PROCEDURE

There are two major aspects to the time share approach:

a. procedures and timing control for switching between different blocks, and

b. reliable circuitry and programming techniques for A/D and D/A transfers.

The switching and timing techniques were developed using track/store (T/S) amplifiers as storage devices instead of the digital computer. Therefore storage could be accomplished "off-line" (not connected to the digital computer).

Two arbitrary coupled linear second order dynamic systems were selected as the problem on which to develop this technique. This choice was motivated by the desire to use a problem whose solution could be obtained analytically. Also, it was important to work with a dynamic system which could be easily implemented "hybrid-wise" to prevent complicating details from obscuring the main purposes. The exact all-analog solution was run concurrently with the various time-share attempts in order to have a visual aid (strip chart output) in determining the success of the procedure.

The equations programmed were:

$$a_1\ddot{x}_1 + b_1\dot{x}_1 + c_1x_1 + d_1x_2 = 0 \qquad (1)$$

$$a_2\ddot{x}_2 + b_2\dot{x}_2 + c_2x_2 + d_2x_1 = 0 \qquad (2)$$

The first step in implementing equations on the analog is to re-scale them to reflect the maximum values of each variable. The equations are thus written in the form:

$$\left(\frac{\ddot{x}_1}{\hat{x}_1}\right) + \left(\frac{\dot{x}_1}{\hat{x}_1}\right)b_1^1 + \left(\frac{x_1}{\hat{x}_1}\right)c_1^1 + \left(\frac{x_2}{\hat{x}_2}\right)d_1^1 = 0 \qquad (3)$$

$$\left(\frac{\ddot{x}_2}{\hat{x}_2}\right) + \left(\frac{\dot{x}_2}{\hat{x}_2}\right)b_2^1 + \left(\frac{x_2}{\hat{x}_2}\right)c_2^1 + \left(\frac{x_1}{\hat{x}_1}\right)d_2^1 = 0 \qquad (4)$$

The circumflex indicates maximum expected value ($\hat{\dot{x}}_1$ = largest value obtained or estimated for $\dot{x}_1$), and

$$b_1^1 = \left(\frac{b_1}{a_1}\frac{\hat{\dot{x}}_1}{\hat{x}_1}\right), \text{ etc.}$$

The scaled variables are those actually solved for on the analog. The constants were chosen to reflect frequencies typically found in cable problems. As implemented, the constants were:

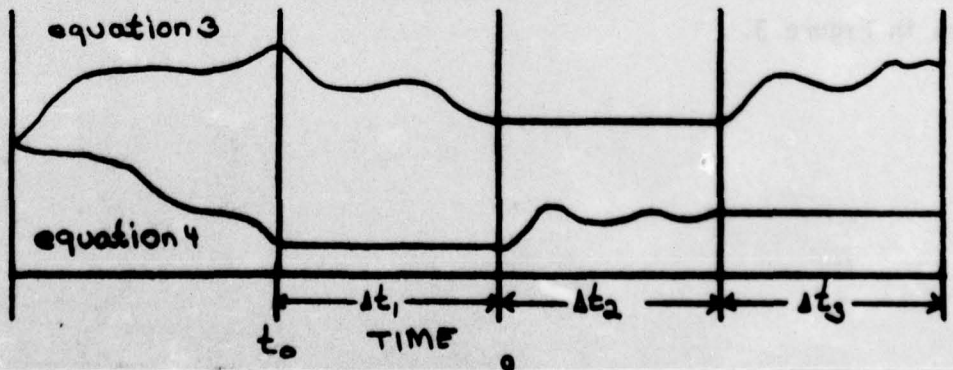
$$b_1^1 = .01,\ c_1^1 = .1,\ d_1^1 = -.05$$

$$b_2^1 = .01,\ c_2^1 = .2,\ d_2^1 = -.05$$

The blocks to be time-shared are the two separate second order systems. This is possible because the same analog circuitry is used to solve equation (3) as is used to solve equation (4). A summary of analog component schematics and mnemonics is given in Appendix A, and Appendix B contains the analog wiring diagram for the all-analog solution. (Figure B1) and the wiring diagram, logic diagram, and timing sequence for the track/store version of the time share procedure (Figures B2, B3, and B4).

Functionally, the component time-sharing technique can be described with the aid of the Figures. Referring to Figure B1, it is recognized that the diagram in Block 1, as far as componentry is concerned, is identical to that in Block 2. However, Block 1 solves equation (3) and Block 2 solves equation (4), with the one difference between the two equations reflected in different settings on digitally controlled attenuator (DCA) 00 (0.1) and DCA 13 (0.2). The implementation shown in Figure B1 gives solutions continuous in time for both Block 1 and Block 2. If discontinuous solutions are acceptable, components can be shared. For example, again referring to Figure B1, suppose that at time t_o the continuous solutions in both Blocks 1 and 2 were frozen and the values of the outputs of integrators 000, 002, 010, and 012 were stored. A continuation of the Block 1 and Block 2 solutions can take place as follows. The stored output of integrator 000 is used as an IC (Initial Condition) on the same integrator. The stored value

of integrator 002 is used as an IC on integrator 002. The stored output of integrator 012 is used as an input to summer 001 through DCA 01 (set at -0.05). The circuit in Block 1 is now run for a short span of time, Δt_1 (Block 1 and Block 2 are disconnected). At the end of the span, Δt_1, the outputs from integrators 000 and 002 are stored in place of the previously stored values. Now using the same components as shown in Block 1 with DCA's 00, 01, 02, 03, and 10 set to the values for the corresponding DCA's in Block 2, equation (4) (Block 2) is solved. With the previously stored output of integrator 010 as an IC on integrator 000, the previously stored output of integrator 012 as an IC on integrator 002, and the most recently stored output of integrator 002 as an input to summer 001 through DCA 01 (set to the value of DCA 12), the circuit in Block 1 is run for another short span of time, Δt_2. At the end of Δt_2 the outputs of the integrators 000 and 002 are stored in place of previously stored values for integrators 010 and 012, respectively. Thus equation (4) has been solved for the small time span, Δt_2. Obviously, equation (3) has not been solved during Δt_2, but has held its solution (obtained at the end of Δt_1) constant over the time span Δt_2. The discontinuous solutions continue with the solving of equation (3) over a time span Δt_3 as follows. With the output of integrator 000 stored at the end of Δt_1 used as an IC on integrator 000, the stored output (at the end of Δt_1) of integrator 002 used as IC on integrator 002, and the value stored in integrator 012 as an input to summer 001 through DCA 01 (set to the value for DCA POT 01) the circuit in Block 1 is run for another short span of time, Δt_3. Again, during Δt_3, equation (4) has not been solved, but has held constant the solution obtained at the end of Δt_2. The sketch below summarizes this technique.



The solutions obtained by the component time-sharing techniques are not the same as the solutions obtained continuously; however, with the proper choice of frame time, the differences can be reduced to an acceptable level.

A more detailed description of the track/store storage method is given in Appendix B.

The next stage in the development of the component time sharing technique was to replace the track/store (T/S) summers with digital-to-analog (D/A) and analog-to-digital (A/D) converters. The circuit is shown in Figure B5. The logic diagram is the same as for the T/S method (Figure B3), but in place of an analog push button starting the run, the digital computer sets a control line high to start the run.

In the diagram of Figure B5, the Track/Store summers (shown in Figure B2) are replaced by A/D and D/A converters. The integrators are initialized during the "A" state; the problem is run during the "B" state; and the values of the integrators are stored in the digital computer during the "C" state.

The results of the A/D, D/A time share method used to solve the sample problem are shown in Figure 1 along with the continuous solution.

HYBRID IMPLEMENTATION OF FOUR NODE BUOY RELEASE

The final phase of the feasibility study was the implementation of a sample problem on the hybrid computer. The problem chosen was the two-dimensional buoy release problem, where a buoy at the end of an anchored cable is released from its initial position at about 45 degrees and allowed to relax to the vertical position. The geometry for the four node problem is shown in Figure 2. The equations as programmed are given in Figure 3.

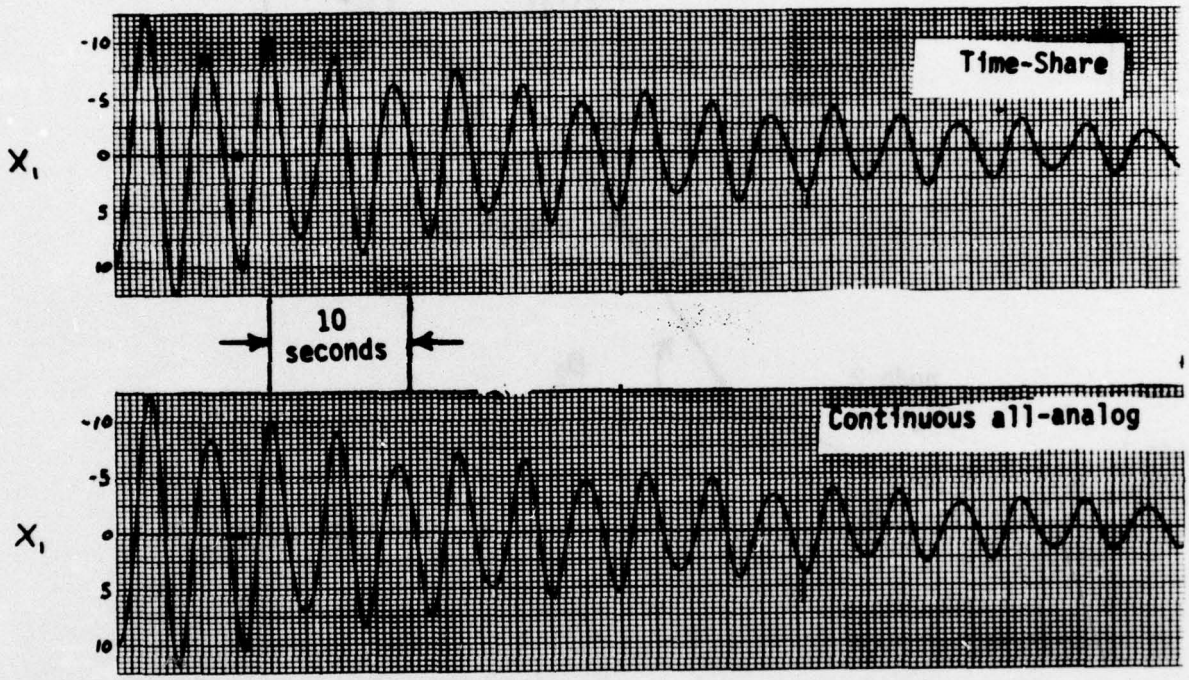
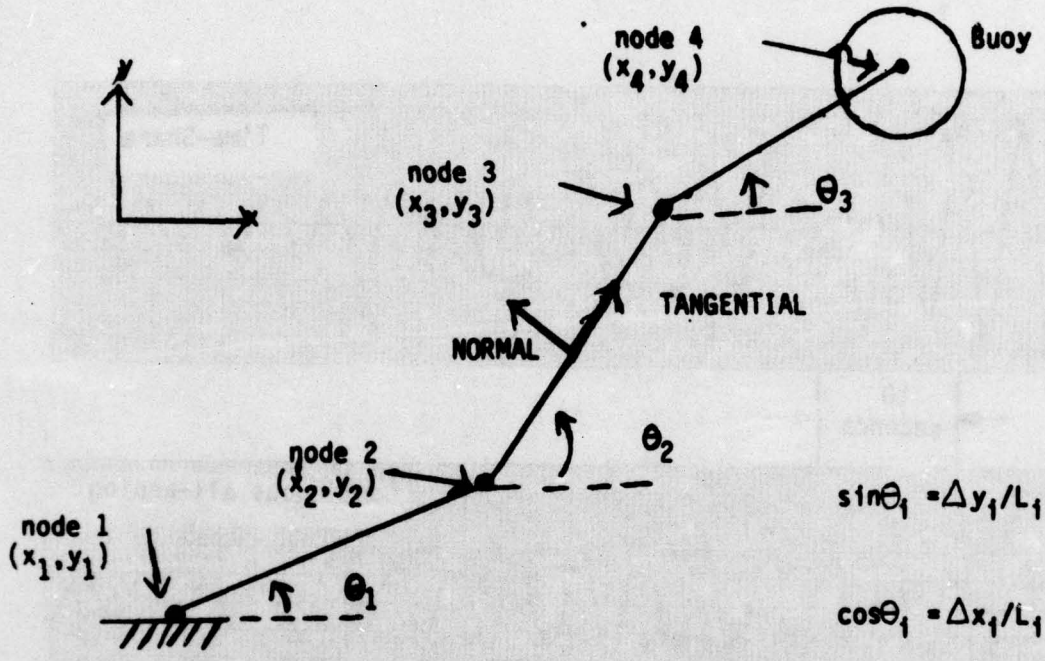


FIGURE 1 - Solution to the Two Coupled Second Order Dynamics Systems Problem



$\sin\theta_i = \Delta y_i / L_i$

$\cos\theta_i = \Delta x_i / L_i$

x - coordinate of node i	x_i
y - coordinate of node i	y_i
x - component of distance between nodes i+1 and i	$\Delta x_i = x_{i+1} - x_i$
y - component of distance between nodes i+1 and i	$\Delta y_i = y_{i+1} - y_i$
distance between nodes i+1 and i	$L_i = (\Delta x_i^2 + \Delta y_i^2)^{1/2}$
strain between nodes i+1 and i	$\Delta L_i = L_i - L_o$

FIGURE 2 - Geometry of Four Node Buoy Release Problem

x-force at node i

$$m\ddot{x}_i = -F^N_{i+} \sin\theta_i - F_{i-} \sin\theta_{i-1} + F^T_{i+} \cos\theta_i + F^T_{i-} \cos\theta_{i-1} + F^x_4 \delta_{i,4}$$

y-force at node i

$$m\ddot{y}_i = F^N_{i+} \cos\theta_i + F^N_{i-} \cos\theta_{i-1} + F^T_{i+} \sin\theta_i + F^T_{i-} \sin\theta_{i-1} + F^y_4 \delta_{i,4} - W_i$$

tangential force on upper/lower element at node i

$$F^T_{i+} = T_i = k \frac{\Delta L_i}{L_o} \left(1 + \tfrac{1}{2} \frac{\Delta L_i}{L_o}\right); \qquad F^T_{i-} = -T_{i-1} = -k \frac{\Delta L_{i-1}}{L_o} \left(1 + \tfrac{1}{2} \frac{\Delta L_{i-1}}{L_o}\right)$$

normal force on upper/lower element at node i

$$F^N_{i+} = F^{N,AM}_{i+} + F^{N,DRAG}_{i+} \quad ; \qquad F^N_{i-} = F^{N,AM}_{i-} + F^{N,DRAG}_{i-1}$$

normal added mass force

$$F^{N,AM}_{i+} = -C_i A^N_{i+} \quad ; \qquad F^{N,AM}_{i-} = -C_i A^N_{i-}$$

acceleration of upper/lower element at node i

$$A^N_{i+} = -\ddot{x}_i \sin\theta_i + \ddot{y}_i \cos\theta_i \quad ; \qquad A^N_{i-} = -\ddot{x}_i \sin\theta_{i-1} + \ddot{y}_i \cos\theta_{i-1}$$

normal drag force on upper/lower element

$$F^{N,DRAG}_{i+} = -\frac{C_d L_o}{2} V^N_{i+} \left|V^N_{i+}\right| \quad ; \qquad F^{N,DRAG}_{i-} - \frac{C_d L_o}{2} V^N_{i-} \left|V^N_{i-}\right|$$

normal velocity of upper/lower element

$$V^N_{i+} = -\dot{x}_i \sin\theta_i + \dot{y}_i \cos\theta_i \quad ; \qquad V^N_{i-} = -\dot{x}_i \sin\theta_{i-1} + \dot{y}_i \cos\theta_{i-1}$$

x,y drag force on node 4

$$F^x_4 = -C_B \dot{x}_4 \left|\dot{x}_4\right| \quad ; \qquad F^y_4 = -C_B \dot{y}_4 \left|\dot{y}_4\right|$$

unstretched distance between nodes	$L_o = 6'/3 = 2'$
tension coefficient	$k = 2.36$
lumped mass of node i	$m_i = .522 \times 10^{-3}$ $i = 1,2,3$; $m_4 = .487 \times 10^{-4}$
total weight in water of node i	$W_i = .11 \times 10^{-2}$ $i = 1,2,3$; $W_4 = -.12845$
added mass coefficient of node i	$C_i = .2439 \times 10^{-3}$ $i = 1,2,3$; $C_4 = .26544 \times 10^{-2}$
drag coefficient of node i	$C_d = .01578$ $i = 1,2,3$; $C_B = -.0102$

FIGURE 3 - Equations of Motion for Four Node Buoy Release Problem

The following simplifications to the equations representing the dynamics of the buoy-cable system were made for the ease of programming. These simplifications were made for this test case only to reduce the number of nonlinear analog components required. They are not inherent limitations of the hybrid approach and would not be made in a normal simulation.

a. tangential drag was neglected

b. a constant normal drag coefficient was used in place of a drag coefficient as a function of Reynolds number

c. in computing the normal drag, the unstretched length of the cable was used rather than the stretched length.

The first stage of the hybrid implementation was to program an all-analog version of the problem. This was done to develop the analog circuitry and analog scaling for the problem. In addition, the HOI analog setup and checkout program was written, which could then be easily modified to give the HOI program for the hybrid implementation.

The second stage was to develop the time share version and logic panel using track/store amplifiers as storage devices, so that the procedure could be debugged "off-line".

The last stage was to change the storage mechanism to the digital, which required replacing the track/store amplifiers with D/A and A/D converters and writing the digital executive program to control real time and to perform the I/O interfacing (D/A and A/D).

One very important feature was discovered during this last stage. A problem with drift and rounding error existed in using the A/D, D/A loop. However, placing a T/S amplifier between all integrators and A/D converters eliminated the problem of drift prior to the A/D conversion. In addition, the sample-and-hold amplifiers in the A/D converters were locked into the sample mode. With this latter precaution, the results obtained with A/D and D/A component time sharing technique agreed exactly with the results obtained with the T/S component time sharing technique.

The problem was implemented as follows: node 1 is anchored (thus has no differential equations)and node 4, (the buoy) is a boundary point and is left on the analog computer to run continuously. The two intermediate nodes, 2 and 3, are involved in the switching. The timing is such that node 4 is run in real time, and nodes 2 and 3 are each run twice real time for a duration of half the time step, yielding a frame time equivalent to that of node 4. The overall timing sequence is:

a. "A" state (analog IC): D/A transfer for initial conditions,

b. "B" state (analog operate): integration at real time for node 4, twice real time for node 2 or 3, and

c. "C" state (analog HOLD): A/D transfer of final values to be used as initial values the next time this node is run.

Figure 4 shows time histories of the x and y displacements for nodes 2, 3, and 4 for the hybrid time-sharing solution to the buoy release cable problem. The dots superimposed on the hybrid results are the results obtained by using an all-digital computer program. Appendix C contains the wiring diagrams and digital computer programs which make up the hybrid simulation, and a listing of the HOI program used to check out the hybrid solution.

Appendix C, page C-2, shows the analog wiring for the integrators and the switching procedure. The wiring is set up so that either the T/S or the A/D storage method may be used. Function relays are used to switch the initial condition inputs from T/S amplifier to D/A converter.

It was found that by setting "A" to .003 seconds, "B" to .02 seconds, and "C" to .002 seconds, representing a node frame time of .025 seconds, the solutions agreed with the all-digital solution based upon a node frame time of .01 seconds.

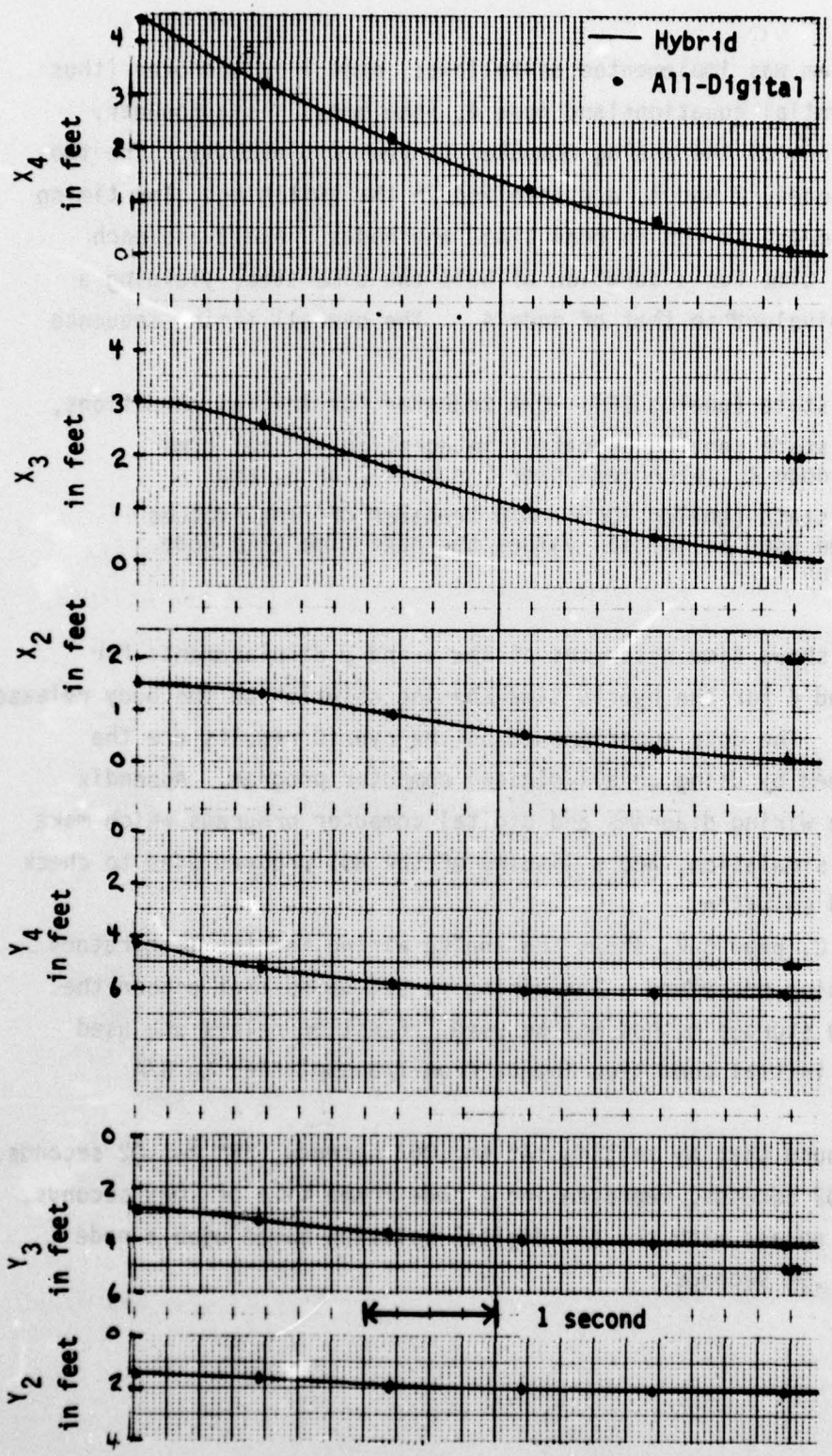


FIGURE 4 - Time Histories of Cable Node Displacements

RECOMMENDATIONS

Included below are several specific recommendations for design and technique modifications to the hybrid program which will result in improved time and accuracy and which can be easily implemented.

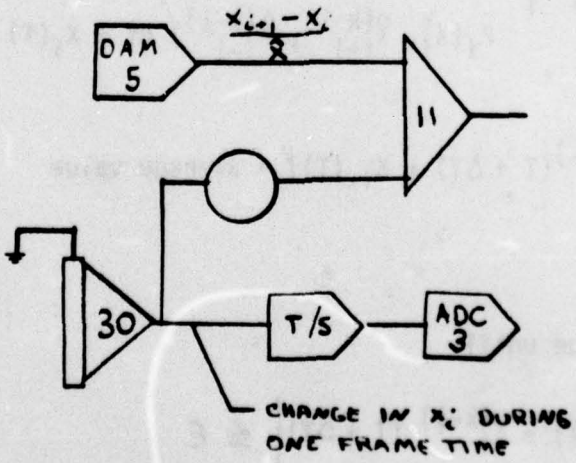
1. Analog-Digital I/O Interface.

There are two types of hybrid I/O interfacing for A/D and D/A conversion, (a) a direct Input/Output interface (DIO) and (b) a PORT interface. The DIO requires the Central Processing Unit (CPU) to execute instructions for each word transferred, whereas the PORT requires the CPU only for the initial transfer. Thus in programs where the frametime is limited by the hybrid I/O it is often possible to significantly increase the throughput by replacing the DIO with the PORT.

The present program uses the DIO interface because the PORT software was not available at the time of this development. It is recommended that in future studies the program be converted to use the PORT interface.

2. Change of Position Variables Stored.

It is recommended that the circuitry shown below be investigated to replace storage of the actual node coordinates with storage of the change in node coordinates.



Making use of this circuitry instead of the circuitry that was used in the hybrid implementation described in this report will result in the elimination of one D/A transfer per degree of freedom and the elimination of one gain of 10 on amplifiers 11, 42, 3, and 31, thus reducing the A/D conversion error by an order of magnitude.

3. Addition of Iterative Scheme to Improve Accuracy.

The procedure used in this report was one in which the position of neighboring nodes were held constant while the motion of the node of interest was integrated. Accuracy was obtained by decreasing the time step until the changes during Δt were small enough to yield the desired results.

A better procedure would be to include an iterative scheme, where several passes are made each time step to obtain self consistent results. The accuracy would be determined by adjustable convergence criteria. One such scheme is the following:

Suppose we have a set of N simultaneous differential equations of the form:

$$X_i = F_i\,(X_i,\ X_{i+1},\ X_{i-1}) \quad X = 1,\ 2,\ \ldots,\ N$$

The equations are "relaxed" to solve,

$$x_i^{(k)} = \int_T^{T+\ T} F_i(X_i^k,\ \hat{X}_{i+1}^{(k-1)},\ \hat{X}_{i-1}^{(k-1)})\ dt + X_i(T)$$

where,

$$\hat{X}_i^{(k)} = \tfrac{1}{2}\,(X_i^{(k)}(T + \Delta T) + X_i\,(T)) = \text{average value}$$

$$x_i^{(k)} = X_i(T)$$

Iterations continue until

$$\left| x_i^{(k)}\ (T + \Delta T) - x_i^{(k-1)}\ (T + \Delta T) \right| \leq \epsilon$$

The value of ϵ should be a value realizable with analog components. No changes in the analog or logic patch panel would be required to implement this more general technique. The only changes required would be some additional coding in the digital program.

DISCUSSION OF VERSATILITY

An expansion from 2 to 3 dimensions would be very straight forward. The procedures would be the same, but there would be more equations (z_i as well as x_i and y_i) and more terms in the existing equations. For example, L_{i+} becomes $\left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2 \right]^{\frac{1}{2}}$. It is possible that fewer nodes will fit in a block and more blocks will be needed per time step, thus requiring more computer time to solve the problem, (not as in the all-digital solution). The concepts, however, remain the same and no new techniques would be required.

Parametric studies can be done fairly easily with hybrid programs. The program is designed such that changing parameters from run to run can be easily accomplished from an input terminal either directly or from a file.

Design changes would require more effort. The initial work in general structure, timing, I/O procedures, etc., would remain the same, but any reprogramming on the analog would require development time, dictated by the nature of the change. The analog could be programmed to handle more general problems initially, but at the sacrifice of analog components.

Procedures involved in expanding to more complex problems will depend heavily on the particular configuration. In general, more equation programming would be involved, either on analog or digital computers. For the sample four-node buoy problem, all of the calculation was done on the analog, and the digital was used solely for "storage" of current values of variables. If a problem is very large, it might be advantageous to do some of the calculations on the digital computer. The consideration then is whether the computation time is least with more of the computation on the analog board (thus fewer nodes per block and more blocks per

frame time) or with more computation on the digital (perhaps requiring more iterations for convergence). For example, one extension would be including nodes with more than 2 elements attached (e.g., nets). This would involve reprogramming large portions of the analog panel. It might be decided that the overall efficiency would be increased if the sines and cosines, for instance, where done digitally. These considerations would have to be worked out for each case.

A hybrid program is fairly machine dependent, in that every hybrid facility has different hardware and software. The general capabilities, however, should be similar enough that only a relatively small effort would be required to convert the hybrid program to run at a similar hybrid facility.

CONCLUSIONS

Both the track/store time-share and the hybrid time-share techniques were tested on a cable system represented by four nodes. Results obtained with both techniques for the four node cable system were found to agree with one another and with results obtained from an all-analog as well as an all-digital implementation. Although the test case was small, it is concluded that large sophisticated systems can be successfully and economically simulated in real time via hybrid techniques.

A reasonable estimate of the running times can be predicted from the test problem. An integration interval of .025 seconds problem time was found to be sufficient. If the PORT interface is used, the timing for one node for this integration interval is estimated as follows:

a. 200×10^{-6} seconds are required for integrators to settle and for D/A PORT transfer,

b. 100×10^{-6} seconds are required for operate time, and

c. 200×10^{-6} seconds are required for A/D PORT transfer time.

Thus for each node, 500×10^{-6} seconds would be required to simulate .025 problem seconds. The number of nodes that can be done in real time is therefore $.025/(500 \times 10^{-6}) = 50$ nodes.

Alternately, it would be possible to simulate a cable with 25 nodes in twice real time, or 100 nodes in ½ real time, or 200 nodes in ¼ real time, etc. Thus, on the basis of the test program, it appears that a 100 node cable system can be simulated in real time on the DTNSRDC Hybrid computer for sophisticated dynamic situations.

Assuming that the above conditions are realizable, it is anticipated that hybrid computing technology can have a significant impact on the dynamic simulation of complex cable systems. It is safe to predict that proper exploitation of this powerful technology can result in the transfer of dynamic cable system developments from hardware (expensive) areas to the simulation (inexpensive) areas.

Finally, in answer to the points addressed in the Introduction concerning the relative merits of hybrid simulation versus an all-digital simulation, the following conclusions are drawn.

a. The program development costs for hybrid simulation are greater than for an all-digital simulation.

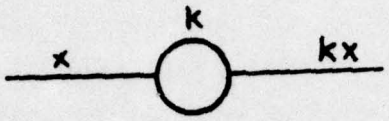
b. Hybrid simulation becomes much more economical than all-digital simulation in performing production runs. Since the hybrid simulation runs real time or faster, the user can afford to simulate many cases which would be prohibitive in cost for an all-digital simulation.

c. Different cable systems may require separate hybrid programs, but the program development costs would be small since the solution technique has already been developed.

There is little question in the minds of the authors that a many node cable dynamics simulation can be implemented using the component time sharing technique together with an iterative scheme. However, it is strongly recommended that a moderately sized (20 node) rather sophisticated dynamic cable system be implemented first; e.g., a cable system towed by a surface ship operating in a seaway. A demonstration of such a successful simulation would remove any uncertainties for continuing on to the most sophisticated cable systems envisioned for the future.

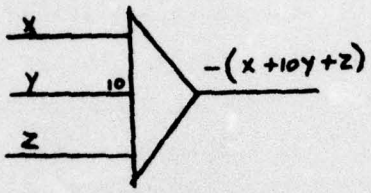
APPENDIX A

DESCRIPTION OF ANALOG COMPONENTS

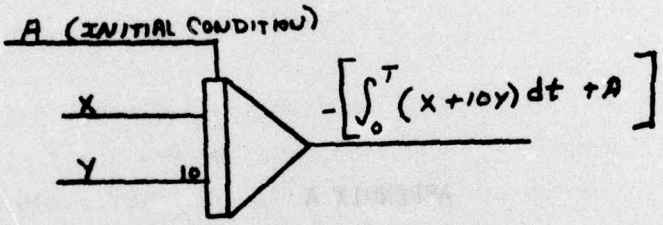


$|k| \leq 1$

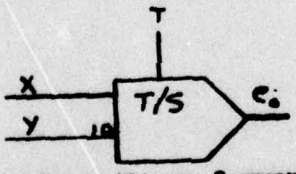
Digitally Controlled Attenuators (DCA)



Summer



Integrator

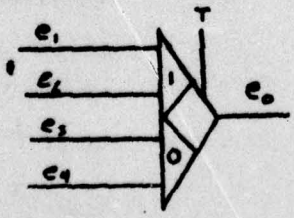


Track/Store Summer

$T=1$, $e_o = -(x+10y)$, TRACK

$T=0$, e_o held at value attained during track at the instant $T=0$ was selected

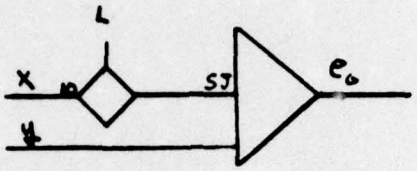
---



$T=1$, $e_o = -(e_1+e_2)$

$T=0$, $e_o = -(e_3+e_4)$

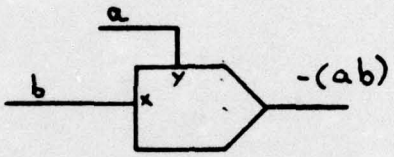
Track/Store Summer Used as an Electronic Switch



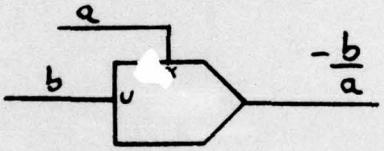
$L=1$, $e_o = -(10x+y)$

$L=0$, $e_o = -y$

Electronic Switch



Multiplier



Divider

$e_o = xD$

where D is digital value transferred to Analog

Digital-to-Analog Multiplier (DAM)

Analog-to-Digital Converter (ADC)

APPENDIX B

DETAILED DESCRIPTION OF TIME-SHARE PROCEDURE

Track and Store Component Time-Sharing Technique

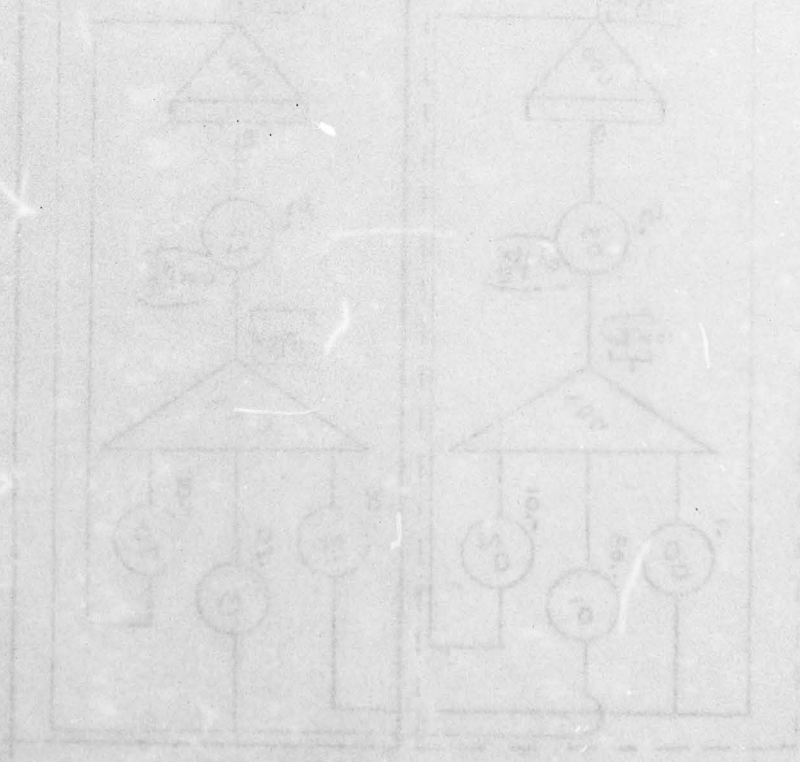
Figure 2 shows the component time-sharing technique using T/S amplifiers. The "1" and "2" states of amplifiers 241, 243, and 251 indicate that the outputs of those amplifiers are the inverses of the state "1" inputs or state "2" inputs. The logic variable "first" (see Figure 3) is used for putting feedbacks around the T/S amplifiers 201, 211, 221, and 231. Without these feedbacks, the T/S summers will drift causing their outputs to be initially non-zero. The variable "first" is also used to initialize integrator 202 during the first two cycles of operation since the T/S amplifiers are initially zero.

Figure 4 shows a timing diagram for the logic variables of Figure 3. That portion of the logic diagram referring to control line 0 is also used for the A/D and D/A component time-sharing technique to be discussed later. When Push Button Flip Flop 0 is depressed the circuit of Figure 1 begins to operate. Depressing this flip flop also starts TIMER 0 (which controls the component time-sharing techniques) cycling between "A", "B", and "C". Flip Flop 31 is initially in the "1" state, consequently integrator 200 is initialized to zero via T/S summer 201. Integrator 002 is initialized via DCA 111. When "B" goes high, integrators 200 and 202 start integrating the equation for state "1" (i.e. Equation 3). While these integrators are in operate, "T_1", remain high and T/S summers 201 and 221 track the output of integrators 200 and 202 respectively.

At the end of "B", "C" goes high forcing T/S summers 201 and 221 to store the value that integrators 200 and 202 respectively had at the end of "B". While "C" remains high the values of integrators 200 and 202 remain at the values they had at the end of "B". At the end of "C", "A" again goes high causing Flip Flop 31 to reset making "2" high. Integrator 200 is initialized zero via T/S summer 211, and integrator 202 is initialized via DCA 111. When "B" goes high, integrators 200 and 202 start integrating the equation for "2" (i. e., Equation 4). While these integrators are on, "T_2" is high and T/S summers 211 and 231 track the output of integrators 200 and 202 respectively. At the end of "B", "C" goes high forcing T/S summers 211 and 231 to store the values that integrators 200 and 202 respectively had at the end

of "B". While "C" remains high, integrators 200 and 202 hold the values they had at the end of "B". When "C" goes low, one complete cycle has been completed. For all remaining cycles, the variable "First" is low, so that the feedbacks around the T/S summers are no longer applied; and the initial condition of integrator 202 will be determined by the state of amplifier 243. This cycle is repeated over and over again as long as Push Button Flip-Flop 0 remains depressed. In the mean time the circuit of Figure B1 continues to operate.

If now output of amplifier 002 (Figure B1) and T/S 221 (Figure B2) differenced and the output of amplifier 012 and T/S 231 are differenced, error terms are generated representing the difference between the continuous solutions and the solutions obtained by the time sharing technique. By varying the "A", "B", and "C" time intervals, it is possible to minimize these errors.



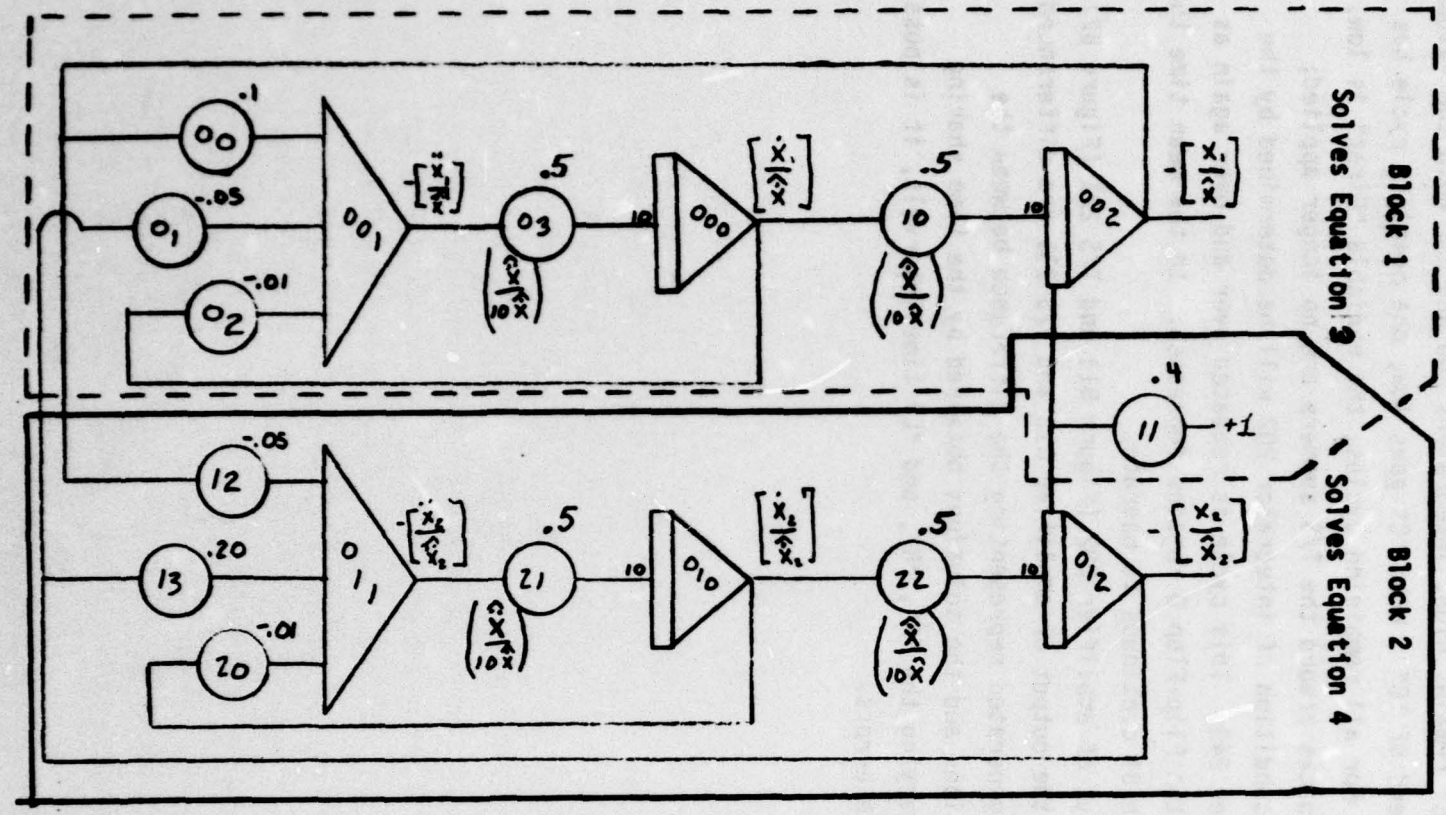


FIGURE B1 - Analog Wiring Diagram for Continuous Analog Solution of Two Coupled Second Order Dynamic Systems

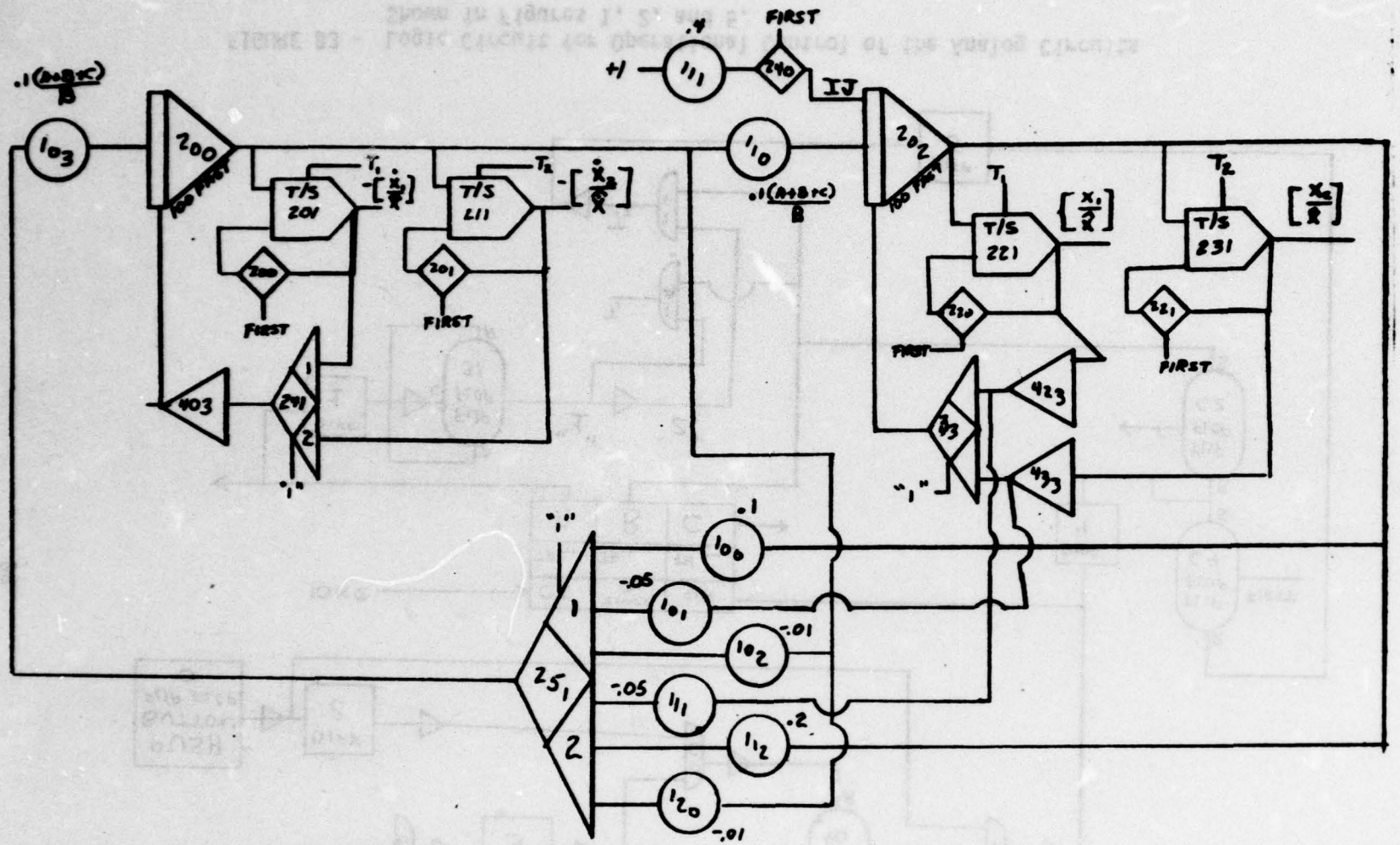


FIGURE B2 - Analog Wiring Diagram for Component-Time-Sharing Technique (Employing Track/Store Summers for Storage) for Solutions to Two Coupled Second Order Dynamic Systems

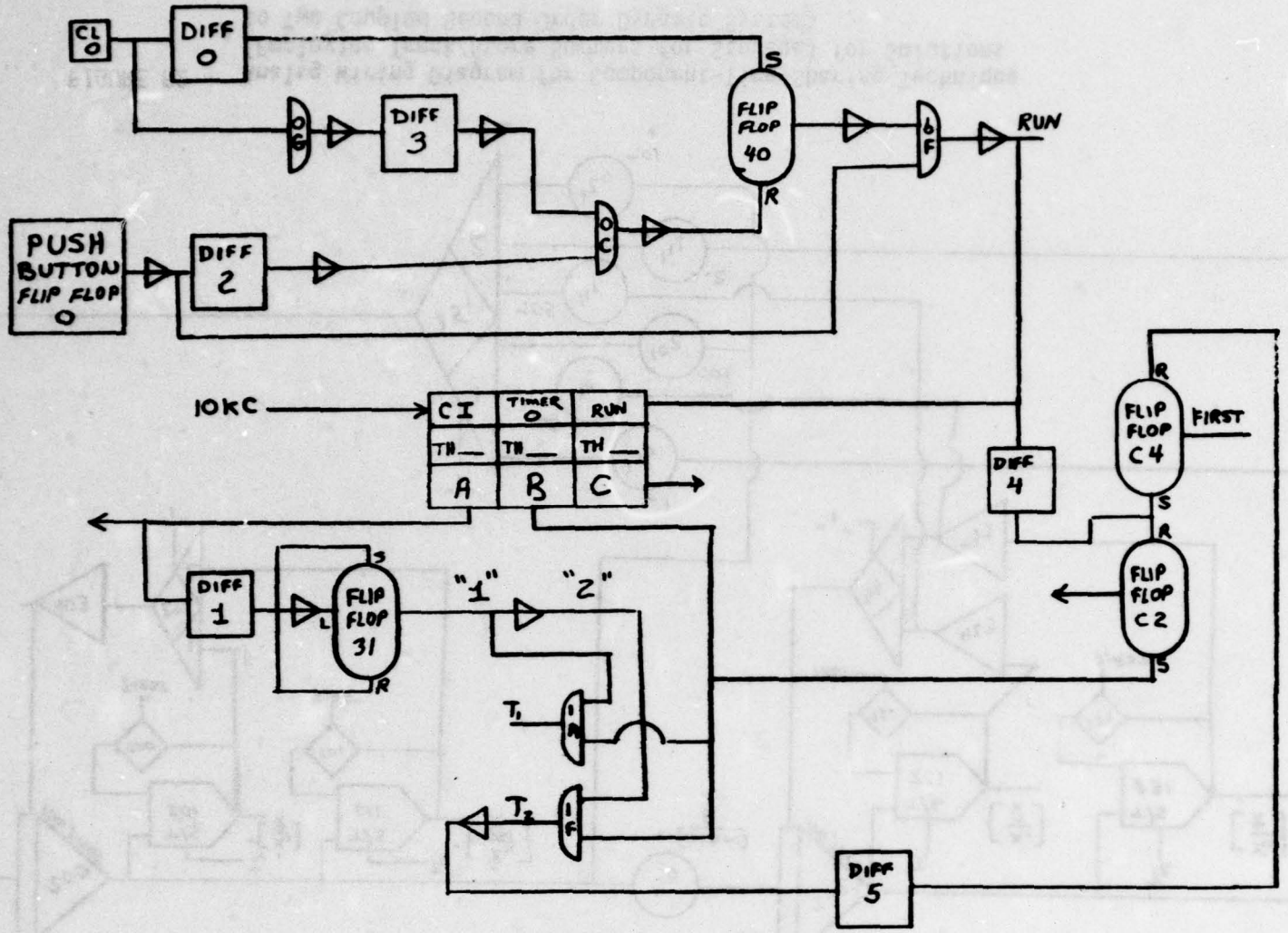


FIGURE B3 - Logic Circuit for Operational Control of the Analog Circuits Shown in Figures 1, 2, and 5.

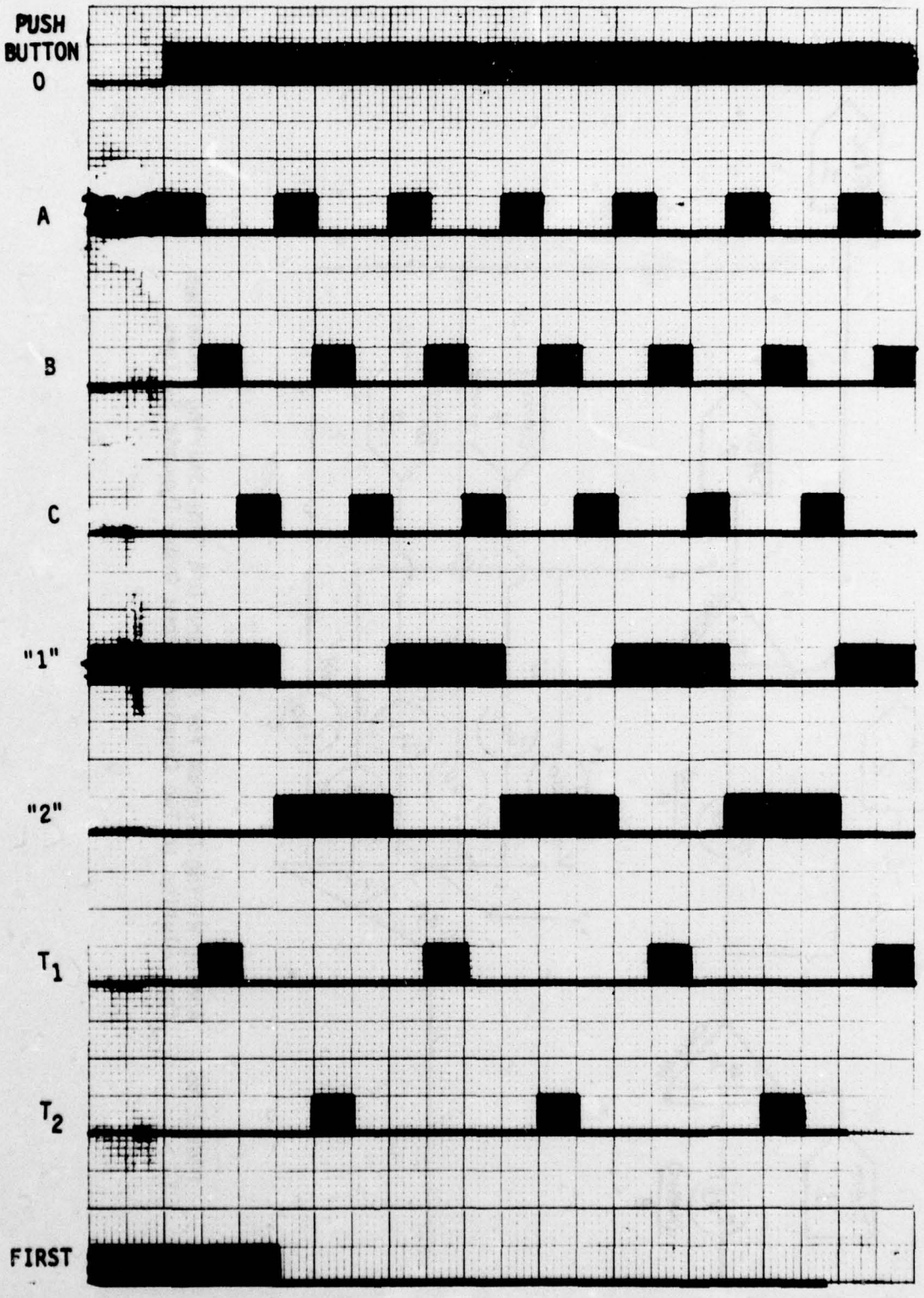


FIGURE B4 - Timing Diagram for Logic of Figure 3.

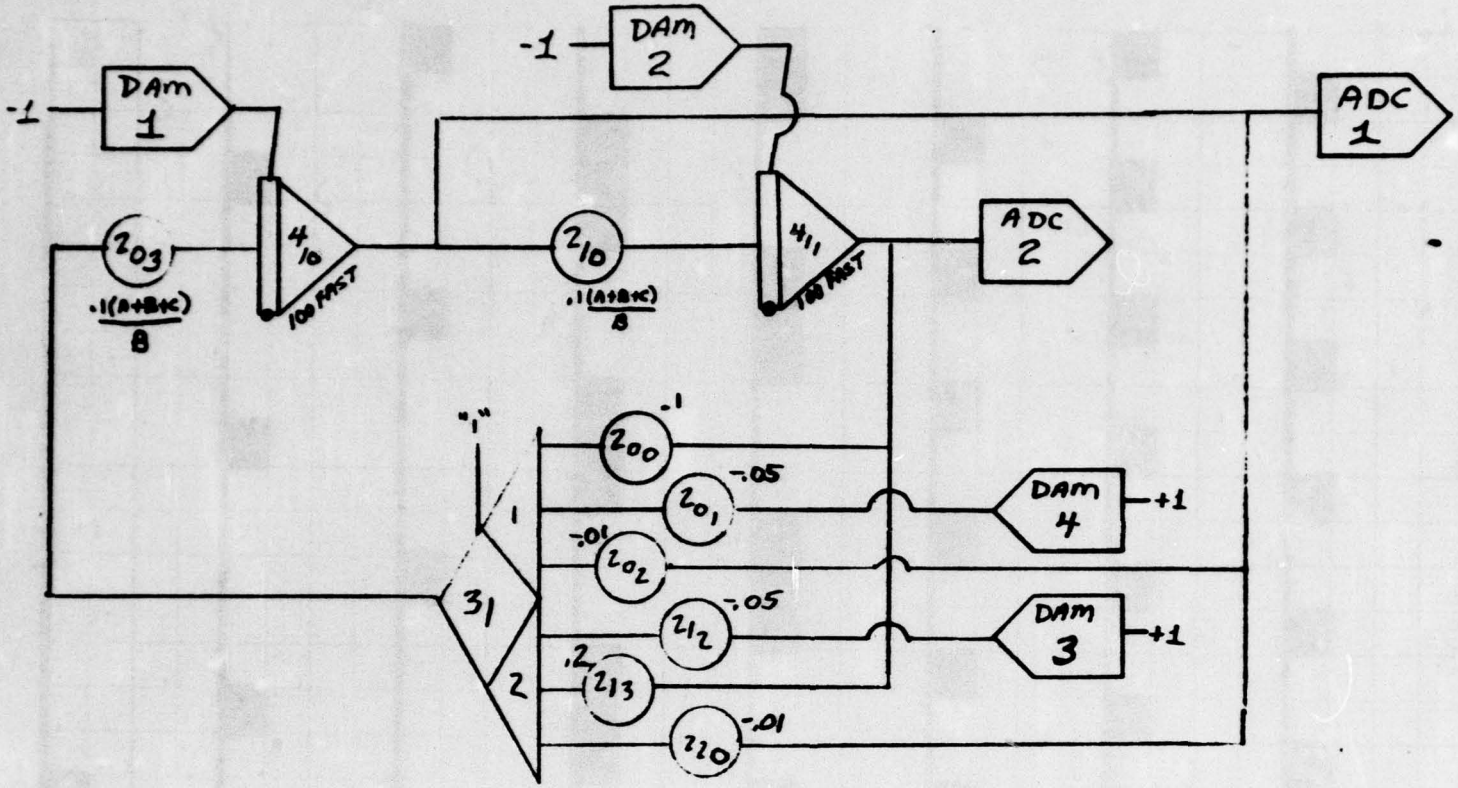
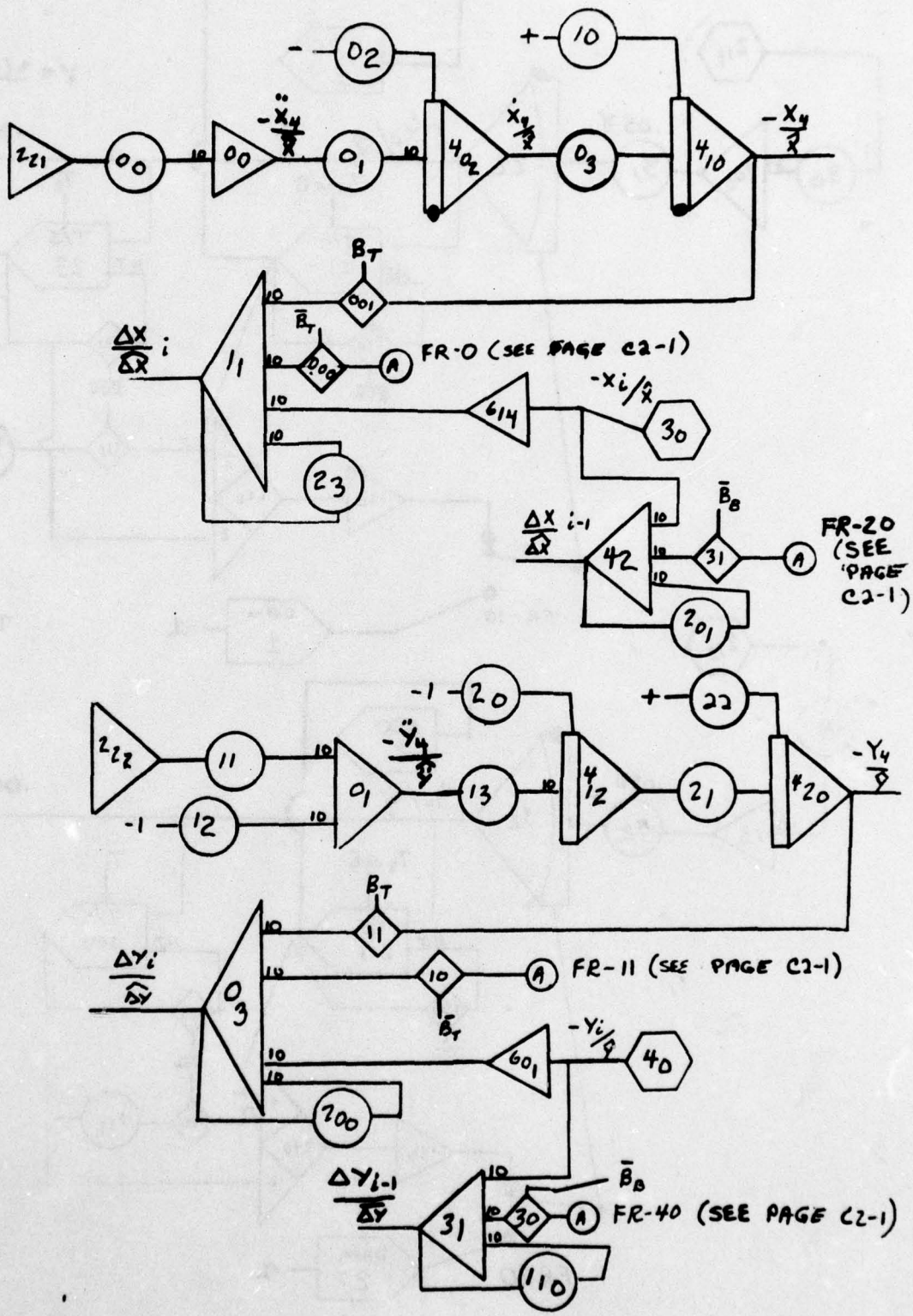


FIGURE B5 - Analog Wiring Diagram for A/D and D/A Time-Sharing Technique for Solution to Two Coupled Second Order Dynamic Systems

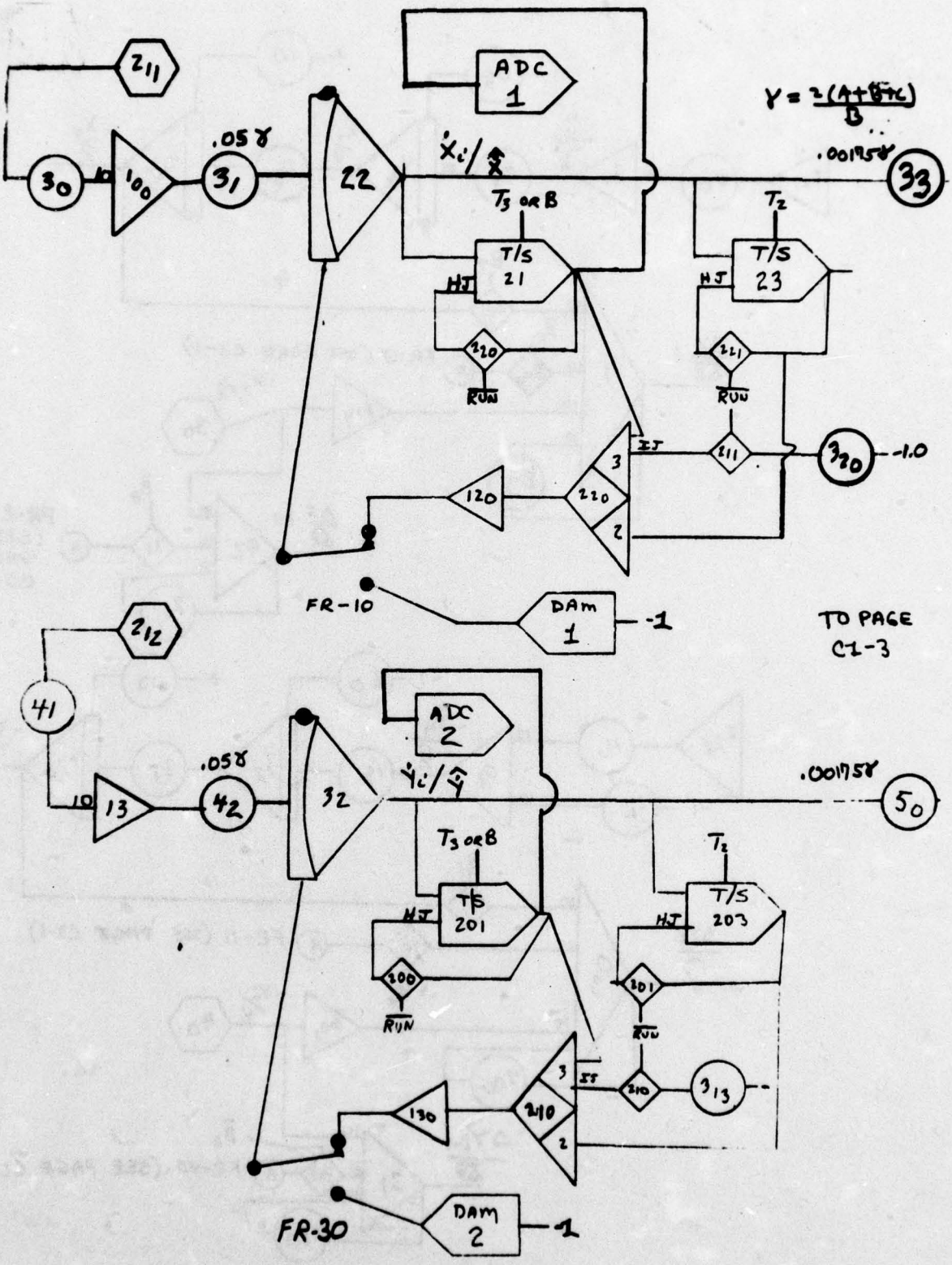
APPENDIX C

HYBRID PROGRAM FOR FOUR NODE CABLE RELEASE PROBLEM

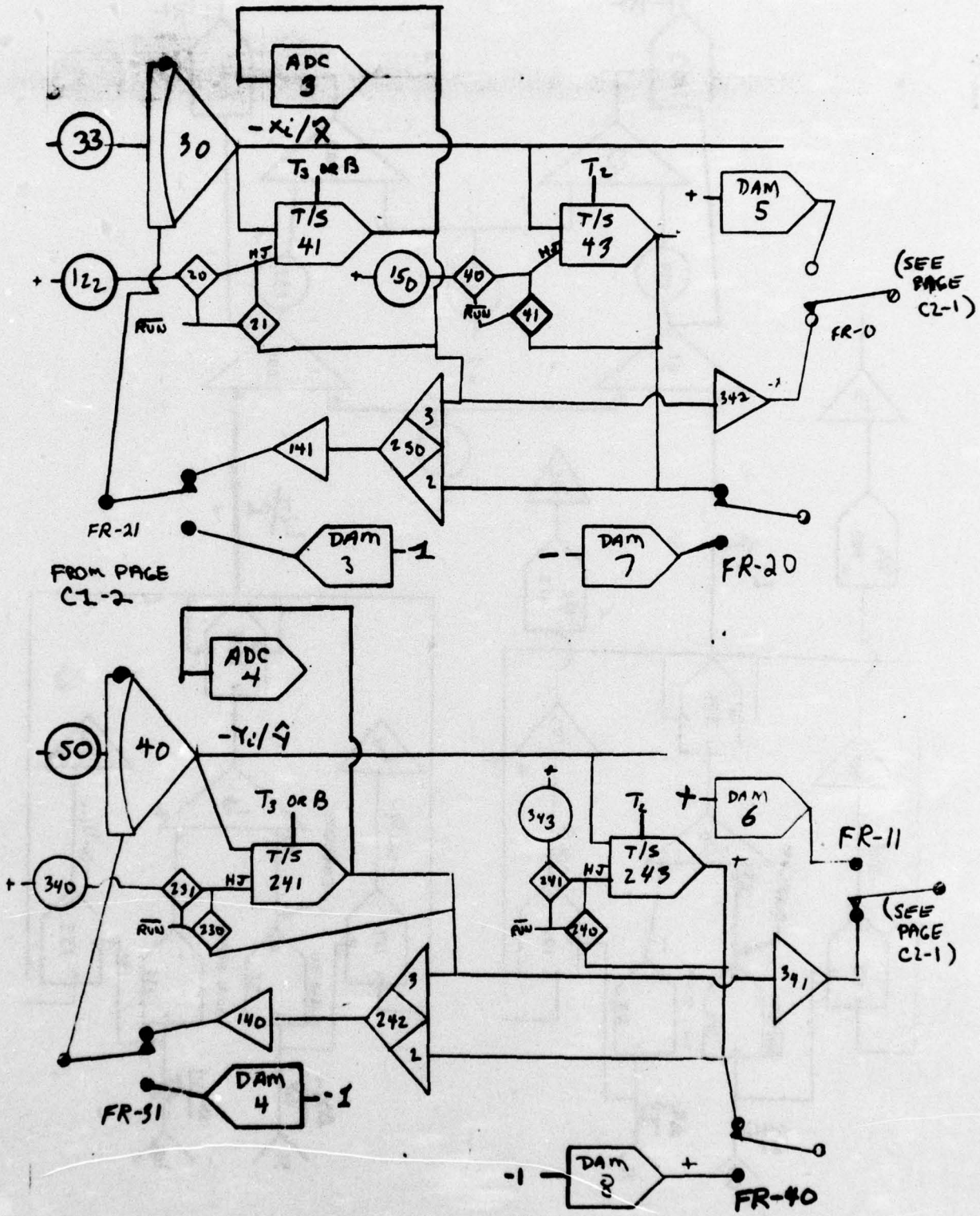
1) Analog Wiring Diagram
2) Logic Panel Diagram
3) Digital Executive Program
4) HOI Setup and Checkout Program



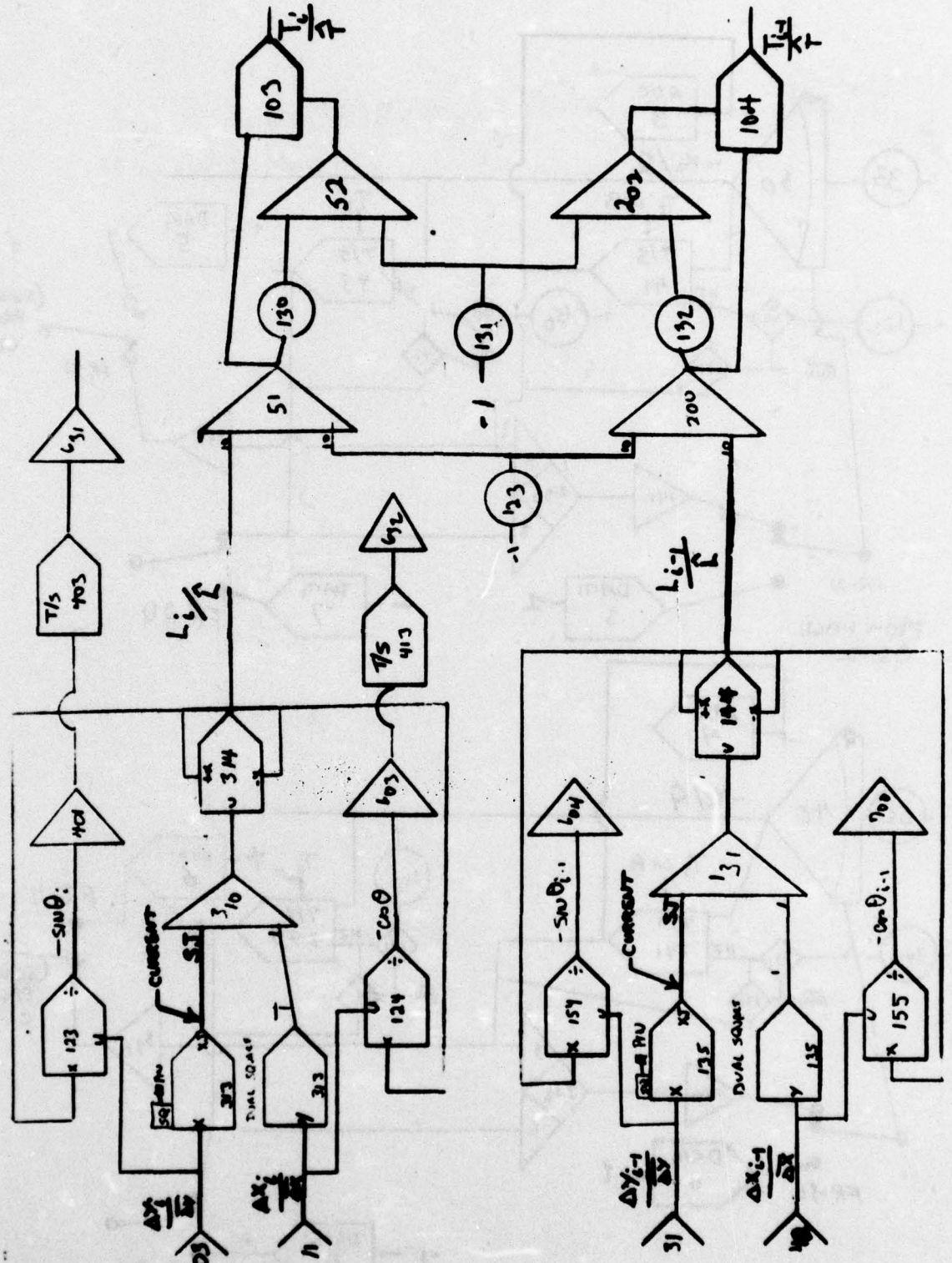
ANALOG WIRING DIAGRAM FOR HYBRID SIMULATION OF
THE FOUR-NODE BUOY RELEASE CABLE PROGRAM



HYBRID WIRING DIAGRAM, CONT.

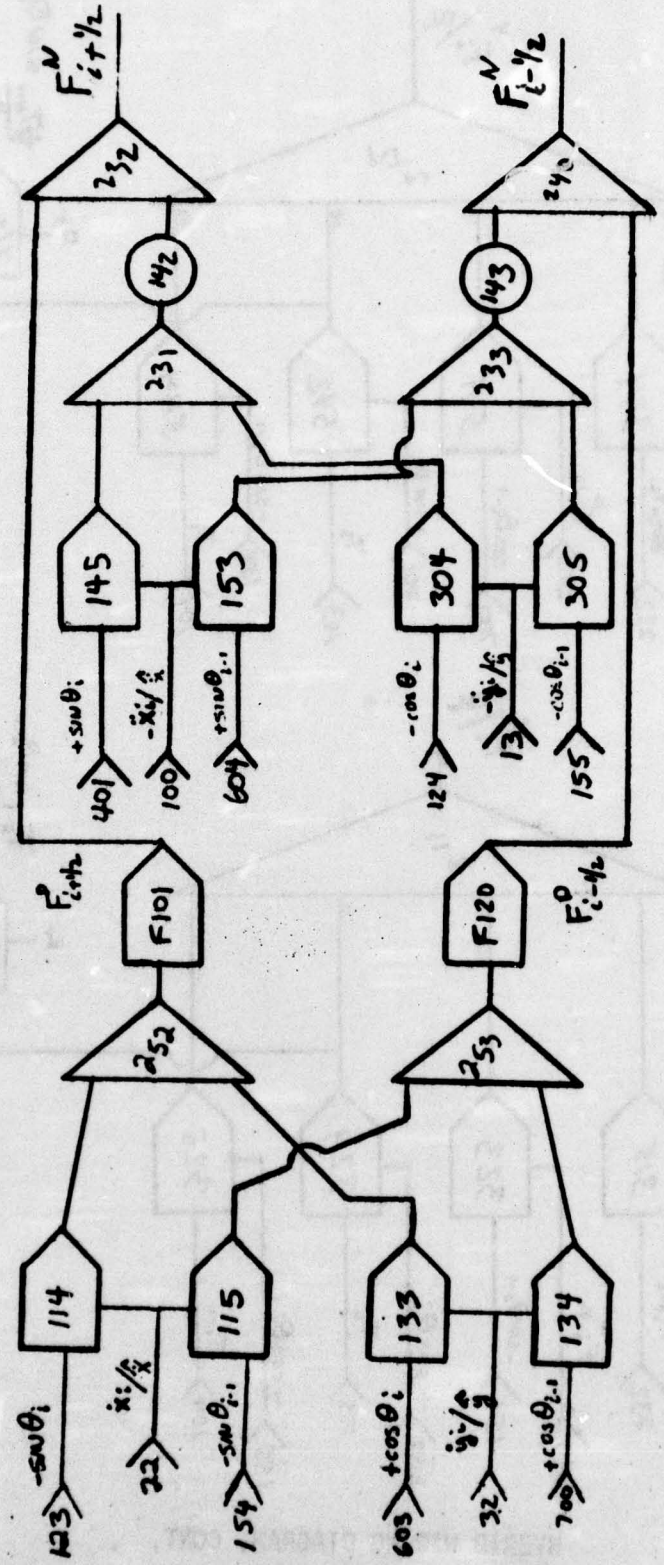


HYBRID WIRING DIAGRAM, CONT.

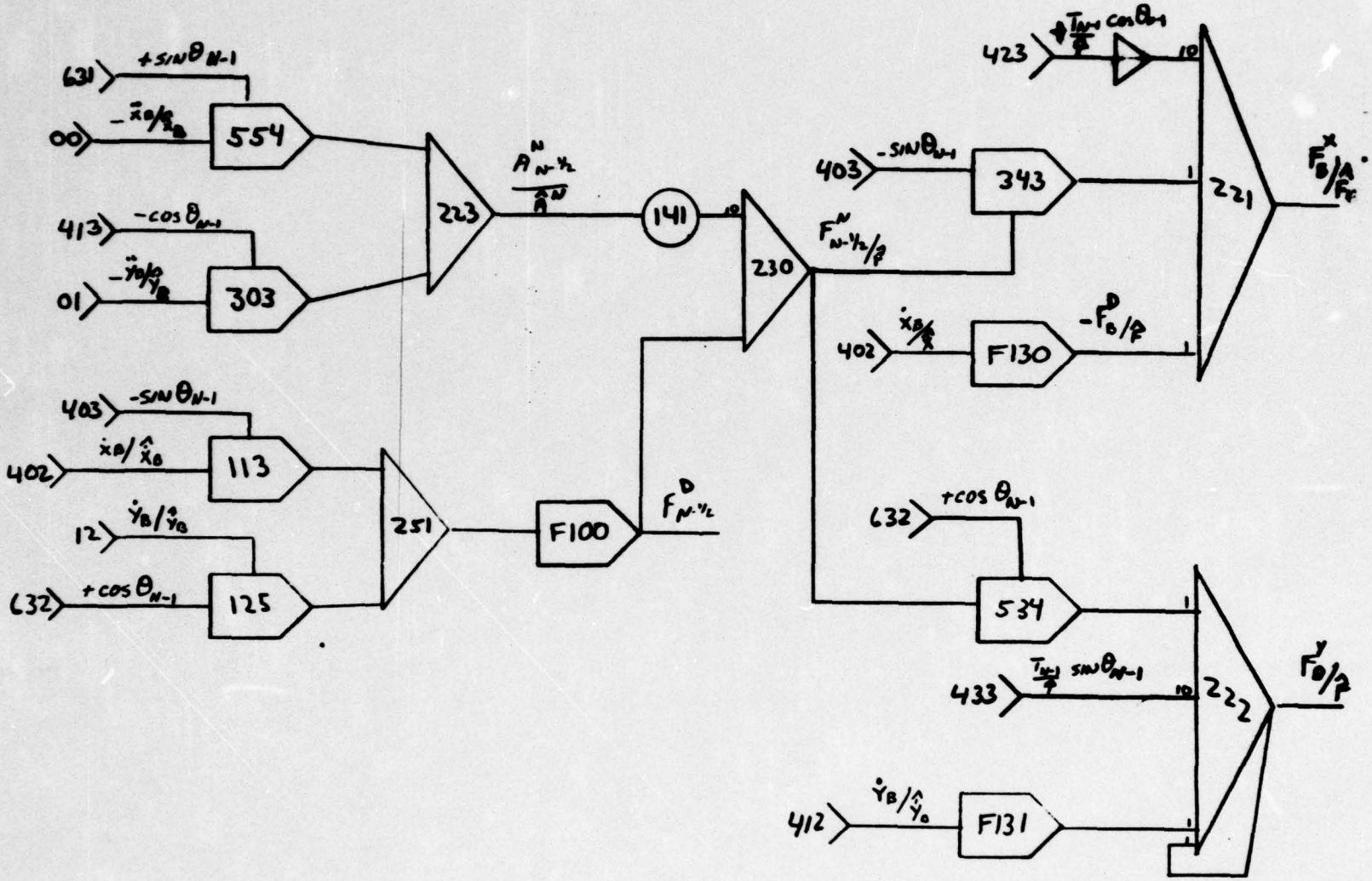


HYBRID WIRING DIAGRAM, CONT.

HYBRID WIRING DIAGRAM, CONT.



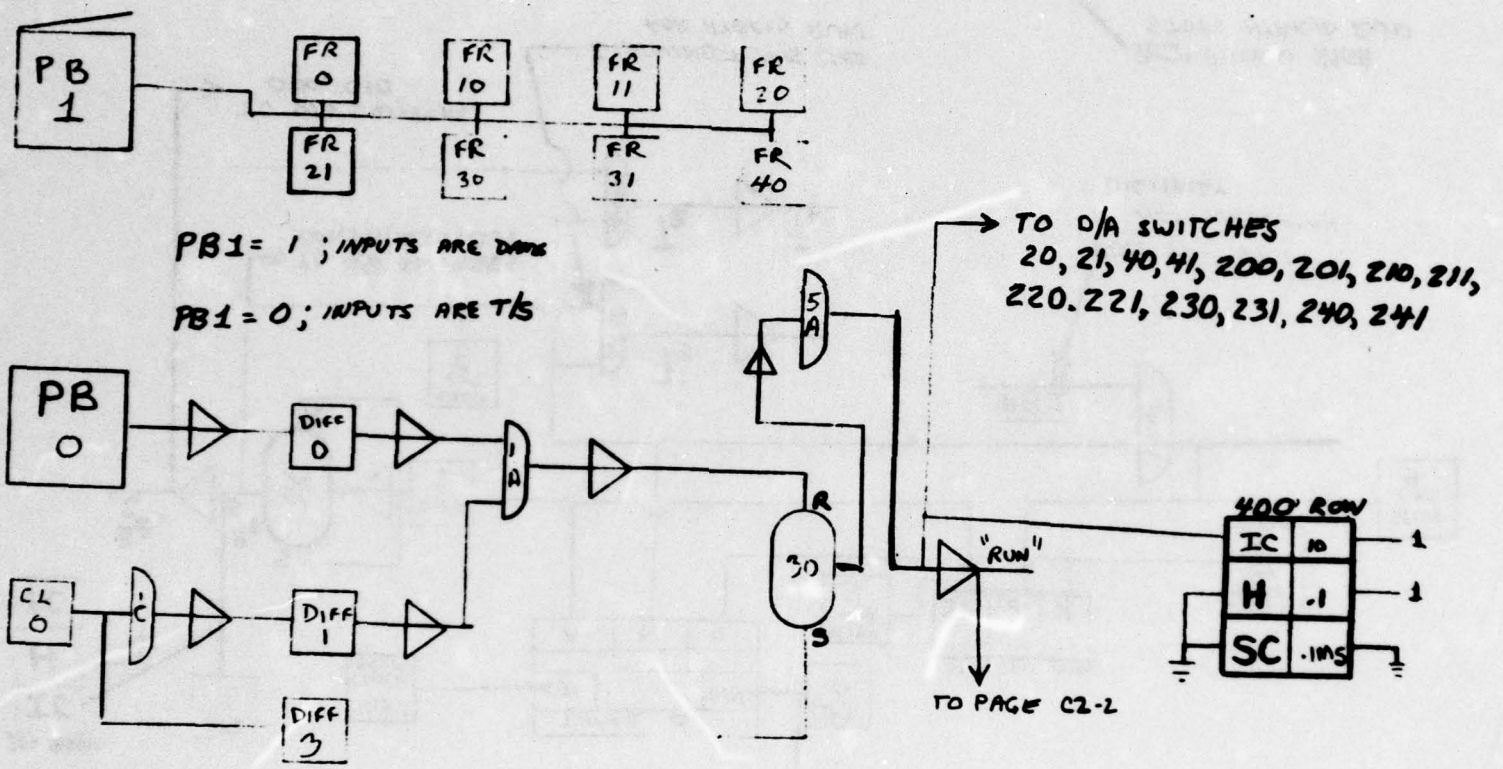
HYBRID WIRING DIAGRAM, CONT.

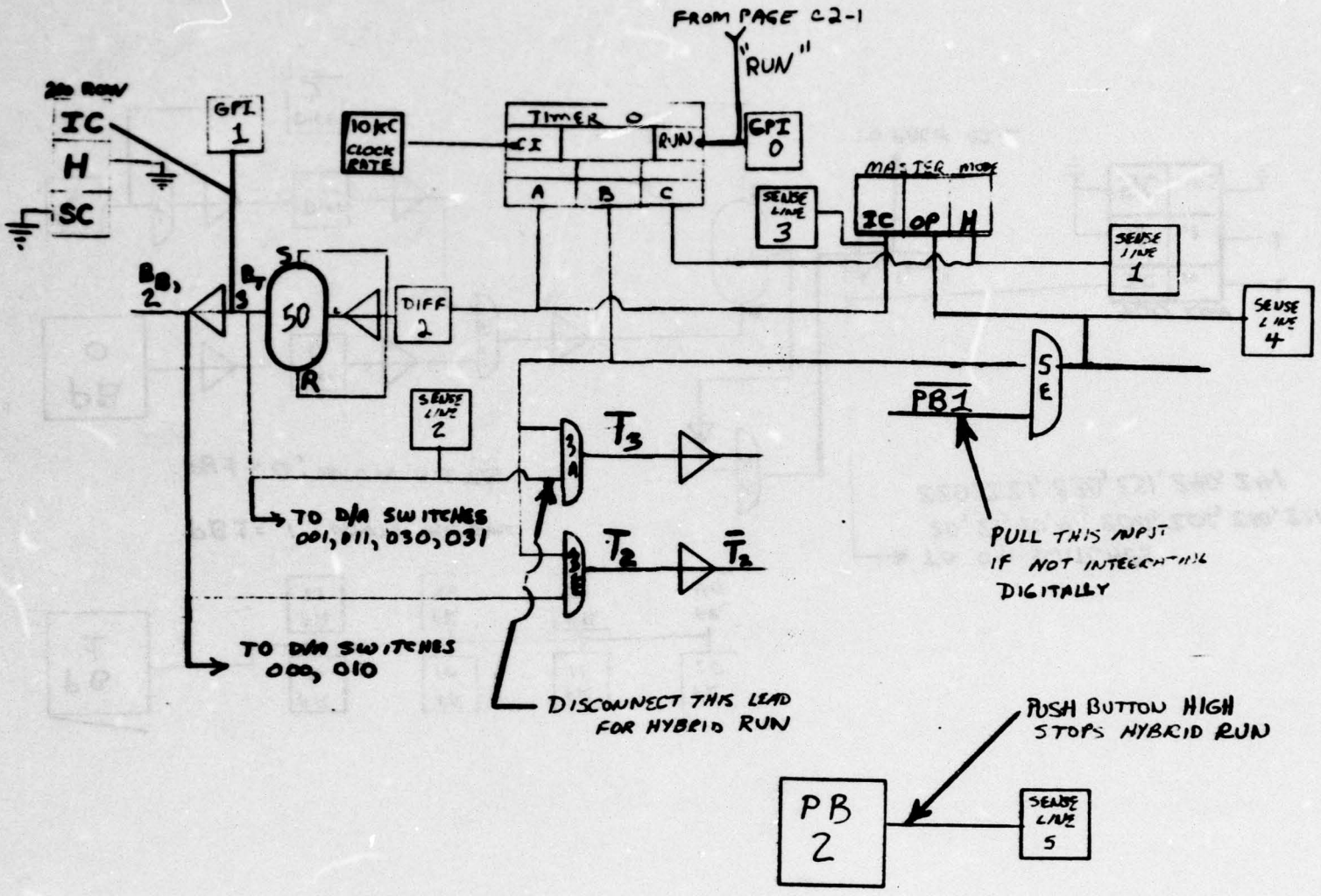


HYBRID WIRING DIAGRAM, CONT.

LOGIC CIRCUIT FOR HYBRID SIMULATION OF FOUR-NODE BUOY RELEASE CABLE PROBLEM

C2-1




FROM PAGE C2-1
"RUN"
2p Rsv
IC
H
SC
GPI 1
10KC CLOCK RATE
TIMER 0
CI
RUN
A
B
C
GPI 0
MASTER MODE
IC
OP
H
SENSE LINE 3
SENSE LINE 1
SENSE LINE 4
50
DIFF 2
SENSE LINE 2
PB1
S
E
T3
T2
T2
TO D/A SWITCHES 001, 011, 030, 031
TO D/A SWITCHES 000, 010
PULL THIS INPUT IF NOT INTERFACING DIGITALLY
DISCONNECT THIS LEAD FOR HYBRID RUN
PUSH BUTTON HIGH STOPS HYBRID RUN
PB 2
SENSE LINE 5

```
 11:25 AUG 11,'76 ID=00CC
JOB CMTLCAB,2157600904,7.     T. MORAN
PCL
COPY CR TO LP
C
C         CL 0 IS RUN
C
C         SENSELINE 1 IS HOLD
C         SENSELINE 2 IS PRESENT NODE.  HIGH IS NODE3, LOW NODE 2
C         SENSELINE 3 IS IC
C         SENSELINE 4 IS OP
C         SENSELINE 5 IS STOP RUN
C
      COMMON /ADCDAM/ ADC,DAM
      DIMENSION X2DOT(10),Y2DOT(10),XDOT(10),YDOT(10),X(10),Y(10)
      DIMENSION DAM(16),ADC(7)
      INTEGER TRSL
      INTEGER CADDR(8)
      DIMENSION SET(8)
      DATA CADDR /001,003,013,021,031,033,042,050/
C
    1 WRITE(101,1001)
      READ(102,1000) ICONS
 1001 FORMAT(1X,'INPUT CONSOLE NO. IN I6')
 1000 FORMAT(616)
      IF(ICONS.NE.1.AND.ICONS.NE.2) GO TO 1
      ITAPE=0
      NTAPE1=0
      NTAPE2=0
      IPORT=0
      IDISK=0
       JCONS1=0
       JCONS2=0
      IF(ICONS.EQ.1) JCONS1=1
      IF(ICONS.EQ.2) JCONS2=1
      CALL RTINIT(ITAPE,NTAPE1,NTAPE2,IPORT,JCONS1,JCONS2,IDISK)
      IPL=1
      CALL  SALN( ICONS , IE)
      IF(IE.NE.0) GO TO 5000
      IPL=2
      CALL SLU(ICONS,IE)
      IF(IE.NE.0) GO TO 5000
      CALL RSCL(0,IE)
      CALL RSCL(1,IE)
      CALL RSCL(2,IE)
      CALL RSCL(3,IE)
      IT3=10
      CALL RTSTART(0)
      IT3=TRSL(3,IE)
      IT3=TRSL(3,IE)
      CALL RTHALT
      WRITE(101,1008) IT3
 1008 FORMAT(1X,'IT3= ',I8)
   10 WRITE(101,1002)
 1002 FORMAT(1X,'NSYS IN I6')
      READ(102,1000) NSYS
      IF(NSYS.NE.4.AND.NSYS.NE.6)  GO TO 10
      WRITE(101,1003) NSYS
 1003 FORMAT(1X,'NSYS= ',I5)
      WRITE(101,1009)
      READ(102,1010) BETA
      WRITE(101,1011) BETA
 1012 FORMAT(1X,'IDIG IN I6. INPUT 1 TO INIG. DIGITALLY' )
      WRITE(101,1012)
      READ(101,1000) IDIG
 1013 FORMAT(1X,'IDIG= ',I6)
      WRITE(101,1013) IDIG
 1009 FORMAT(1X,'TIME SCALE, BETA, IN F10.5')
```

```
 1010 FORMAT(F10.5)
 1011 FORMAT(1X,'BETA = ', F10.5      )
   20 CONTINUE
      CALL TIMEZ(NSYS,X2DOT,Y2DOT,XDOT,YDOT,X,Y)
C
CCCC  LOAD INITIAL DAM VALUES
CCCC
C     ALWAYS START IN STATE 3
      I=3
CCCC  IT2IN=TRSL(2,IE)
      IT2IN=TRSL(2,IE)
      I=2
C     IF(IT2IN.EQ.1) I=3
       DAM(1)=XDOT(I)
       DAM(2)=YDOT(I)
       DAM(3)=X(I)
       DAM(4)=Y(I)
       DAM(5)=X(I+1)
       DAM(6)=Y(I+1)
       DAM(7)=X(I-1)
       DAM(8)=Y(I-1)
       DAM(9)=XDOT(3)
      DAM(10)=XDOT(2)
      DAM(11)=YDOT(3)
      DAM(12)=YDOT(2)
      DAM(13)=X(3)
      DAM(14)=X(2)
      DAM(15)=Y(3)
      DAM(16)=Y(2)
      CALL RTSTART(0)
      CALL LDDA(1,16,0,DAM,IE)
      CALL TLDA(IEDUM)
      CALL RTHALT
      WRITE(101,1004)
 1004 FORMAT(1X,'IC, OP, HOLD, AND CLOCK IN 4I6')
      READ(102,1000) IC,IOP,IHOLD,ICLOCK
      XIC=IC
       OP=IOP
      HOLD=IHOLD
      CLOCK=ICLOCK
      X2DMAX=35.0
      XDMAX=3.5
      XMAX=10.
      XNM2=NSYS-2
      TIMEF=XNM2 + (XIC + OP + HOLD )
      GAMMA=TIMEF/OP
      TIMEF=BETA*TIMEF
      DT1 = TIMEF*X2DMAX/XDMAX/CLOCK
      DT2=TIMEF*XDMAX/XMAX/CLOCK
 1005 FORMAT(1X,'DT1= ',E12.5,'    DT2= ',E12.5)
C     SET DCA'>
      SET(1)=BETA
      SET(3)=SET(1)
      SET(2)=0.35*BETA
      SET(4)=SET(2)
      SET(5)=0.05*GAMMA
      SET(6)=.00175*GAMMA
      SET(7)=SET(5)
      SET(8)=SET(6)
      CALL RTSTART(0)
      IPL=30
      DO 1800 I=1,8
      IADDR=LADDR(I)
      VALUE=SET(I)
      CALL WDCAS(IADDR,0,VALUE,1,IE)
      IF(IE.NE.0) GO TO 5000
 1800 CONTINUE
      CALL RTHALT
      WRITE(101,1000) DT1,DT2
      WRITE(101,1007)
```

```
C    START REAL TIME
 1007 FORMAT(1X,'TYPE 1 TO START. ANYTHING ELSE REINPUTS DATA')
      READ(102,1000) ISTART
      IF(ISTART.NE.1) GO TO 10
      NSM1=NSYS-1
      CALL RTSTART(1)
      IPL=3
      CALL SSCL(0,IE)
      IF(IE.NE.0) GO TO 5000
C
 1500 CONTINUE
      CALL RSCL(1,IE)
      CALL RSCL(2,IE)
      CALL RSCL(3,IE)
      DO 2000 J=2,NSM1
      CALL SSCL(1,IE)
      IPL=4
C     START IN IC
      IT3=IRSL(3,IE)
 1503 IT3=IRSL(3,IE)
      IF(IE.NE.0) GO TO 5000
      IF(IT3.NE.1)   GO TO 1503
      IT2IN=IRSL(2,IE)
      IT2IN=IRSL(2,IE)
      IF(IE.NE.0) GO TO 5000
      I=5*J
      DAM(1)=XDOT(I)
      DAM(2)=YDOT(I)
      DAM(3)=-X(I)
      DAM(4)=-Y(I)
      DAM(5)=-X(I+1)
      DAM(6)=-Y(I+1)
      DAM(7)=-X(I-1)
      DAM(8)=-Y(I-1)
      DAM(9)=XDOT(3)
      DAM(10)=XDOT(4)
      DAM(11)=YDOT(3)
      DAM(12)=YDOT(4)
      DAM(13)=-X(3)
      DAM(14)=-X(2)
      DAM(15)=-Y(3)
      DAM(16)=-Y(2)
C
C   LOAD DAMS
      IPL=5
      CALL LBDA(1,16,0,DAM,IE)
      IF(IE.NE.0) GO TO 5000
C
C    XFER DAMS
      CALL TLDA(IEDUM)
C
C   CHECK THAT STILL IN IC
      IPL=6
      CALL SSCL(2,IE)
      IT3=IRSL(3,IE)
      IT3=IRSL(3,IE)
      IF(IE.NE.0) GO TO 5000
      IF(IT3.NE.1) GO TO 8000
C
      IF(IDIG.NE.1)  IPL=9; GO TO 1502
C   WAIT FOR OP
      IPL=7
      IT4=IRSL(4,IE)
 1501 IT4=IRSL(4,IE)
      IF(IE.NE.0) GO TO 5000
      IF(IT4.NE.1) GO TO 1501
C
      CALL SSCL(3,IE)
C    READ ADC'S
```

```
      IPL=8
      CALL CRBC(1,3,0,ADC,IE)
       IF(IE.NE.0) GO TO 5000
      X2DOT(I)=ADC(1)
      Y2DOT(I)=ADC(2)
C
C    WAIT FOR HOLD
      IPL=9
 1502 IT1=TRSL(1,IE)
      IT1=TRSL(1,IE)
      IF(IE.NE.0) GO TO 5000
       IF(IT1.NE.1) GO TO 1502
      IF(IDIG.NE.1) CALL CRBC(1,7,0,ADC,IE)     ; GO TO 1900
C2000 CONTINUE
C
C    INTEGRATE
C     DO 3000 I=2,NSYS
      Y(I)= Y(I)  +  YDOT(I)*DT2
      X(I)=X(I)  +   XDOT(I)* DT2
      YDOT(I)=YDOT(I)  +  Y2DOT(I) * DT1
      XDOT(I)=XDOT(I)  +  X2DOT(I) * DT2
      IF(IDIG.EQ.1) GO TO 1901
C3000 CONTINUE
 1900 CONTINUE
      XDOT(I)=ADC(3)
      YDOT(I)=ADC(4)
      X(I)= ADC(5)
      Y(I)= ADC(6)
 1901 CONTINUE
C    STILL IN HOLD
      IPL=12
      IT1=0
      IT1=TRSL(1,IE)
      IT1=TRSL(1,IE)
      IF(IE.NE.0) GO TO 5000
C
C   SAME STATE ?
C
      IPL=11
      IT2=TRSL(2,IE)
      IT2=TRSL(2,IE)
      IF(IE.NE.0) GO TO 5000
C
C SAME STATE AND HOLD ?
      IF(IT2.NE.IT2IN.AND.IT1.NE.1) GO TO 8000
 2000 CONTINUE
C
C    TIME TO QUIT ?
      IPL=10
      IT5=TRSL(5,IE)
      IT5=TRSL(5,IE)
      IF(IE.NE.0) GO TO 5000
      IF(IT5.NE.1) GO TO 1500
      CALL RSCL(0,IE)
      CALL RSCL(1,IE)
      CALL RSCL(2,IE)
      CALL RSCL(3,IE)
      CALL RTMALT
   20 WRITE(101,1006)
 1006 FORMAT(1X,'TYPE 1 TO REPEAT'/1X
     1'TYPE 2 TO CHANGE NSYS'/1X,
     2'TYPE 3 TO QUIT')
       READ(102,1000) NEXT
      IF(NEXT.EQ.1) GO TO 20
      IF(NEXT.EQ.2) GO TO 10
      GO TO 6000
 8000 CALL RTMALT
      CALL RSCL(0,IE)
      WRITE(101,8001) IPL
 8001 FORMAT(1X,'TIME FRAME VIOLATION AT ',I5)
```

```
      GO TO 30
 5000 CONTINUE
      CALL RTHALT
      CALL RSCLI(0,I&)
      WRITE(101,5001) IE,IPL
 5001 FORMAT(1X,'ERROR ',I5,'  IN HYB CALL AT PLACE ',I5)
 6000 CONTINUE
      CALL DAMOUT
      CALL RTSTOP
      END
      SUBROUTINE DAMOUT
      COMMON /ADCDAM/ADC,DAM
      DIMENSION ADC(7),DAM(16)
      DO 10 I=1,16
      DAM(I)=0.0
   10 CONTINUE
C
      CALL RTSTART(0)
C    LOAD
      CALL LSDA(1,16,0,DAM,IE)
C    XFER
      CALL TLDA(IE)
       CALL RTHALT
      RETURN
      END
      SUBROUTINE TIME2(N,X2DOT,Y2DOT,XDOT,YDOT,X,Y)
      DIMENSION X2DOT(N),Y2DOT(N),XDOT(N),YDOT(N),X(N),Y(N)
      DIMENSION XIC4(2),YIC4(2),XIC6(4),YIC6(4)
      DATA XIC4 /.15004,.29895/
      DATA YIC4 /.19415,.28955/
      DATA XIC6/ 0.,0.,0.,0./
      DATA YIC6/ 0.,0.,0.,0./
      NSM1=N-1
       DO 10 I=2,NSM1
      X2DOT(I)=0.0
      Y2DOT(I)=0.0
      XDOT(I)=0.0
      YDOT(I)=0.0
   10 CONTINUE
      XDOT(1)=0.0
      YDOT(1)=0.0
      XDOT(N)=0.0
      YDOT(N)=0.0
      Y(1)=0.0
      X(1)=0.0
      Y(N)=0.0
      X(N)=0.0
      IF(N.EQ.6) GO TO 20
      DO 15 I=2,NSM1
      X(I)=XIC4(I-1)
      Y(I)=YIC4(I-1)
   15 CONTINUE
      RETURN
   20 DO 25 I=2,NSM1
      X(I)=XIC6(I-1)
      Y(I)=YIC6(I-1)
   25 CONTINUE
      RETURN
C    !!!   ???JUU???
C    !!!
      ENTRY INT62
      ENTRY INT63
      ENTRY INT64
      ENTRY INT65
      ENTRY INT66
      ENTRY INT67
      ENTRY INT68
      ENTRY INT69
      ENTRY INT6A
```

```
      ENTRY  INT6B
      ENTRY  INT6C
      ENTRY  INT6D
      ENTRY  INT6E
      ENTRY  INT6F
      ENTRY  INT70
      ENTRY  INT71
      ENTRY  INT72
      ENTRY  INT73
      ENTRY  INT74
      ENTRY  INT75
      ENTRY  INT76
      ENTRY  INT77
      ENTRY  INT78
      ENTRY  INT79
      ENTRY  INT7A
      ENTRY  INT7B
      ENTRY  INT7C
      ENTRY  INT7D
      ENTRY  INT7E
      RETURN
C     !!!
      END
C!-OAP (LMN,LABLE),(LEF),(RISYS,RTLIB),(ID),(SYS),(MAIN),(ATIME2)),(
C!-(9100),(USEFT),(LDEF),(PERM),(MAP),(TSS,DOT),(UNSAT),(RTLIB),(IP4),(
```

```
 11:28 AUG 11,'76 ID=00CD
JOB CNTLCAB,2157600404,A.     T. MORAN,  HOI TIME-SHARE CABLE
PCL
C CR TO HOITSCAB
C HOITSCAB TO LP
  "PART 01"!
01.001)       "HOI - CABLE "!
01.002)       "6/08/76"!
01.003)       @IX,@WA,X!   NORMAL!
01.005)       "INPUT CONSOLE #"!  ICONS=0, ICONS=
01.006)       ICONS,USE!   1,C!
01.009)
01.010)       "STATIC TEST?"!  NORMAL!  DO=0, DO=
01.015)       (DO==0)?  1.510.
01.020)       10!  11!  3!
01.030)       "LIST OPTION"  80!
01.040)       "ON-LINE ST SETUP? - YES - 2"!  DO=
01.050)       (DO==2)?  6!
01.509)
01.510)       "INITIAL CONDITIONS?"!  NORMAL!  DO=0,  DO=
01.515)       (DO==0)?  1.998.
01.520)       .10! 12!  3!
01.530)       "LIST OPTION"  80!
01.540)       "ON-LINE IC SETUP? - YES - 2"!  DO=
01.550)       (DO==2)?  6!
01.998)   "PART 01 DONE"!  HALT!
  "PART 03"!
03.001)       "OFF LINE CHECK"
03.005)       NORMAL!  @IX,@WA,X!
03.013)       "DIG CALC"  13!
03.021)       "COEFFS"  21!
03.022)       "DERIVS"  22!
03.023)       "AMPS"  23!
03.025)       "FCN GENS"  25!
03.031)       "POTS"  31!
03.090)
03.100)       "OFF-LINE VERIFY?"!  IV=0, IV=
03.101)       (IV==0)?  3.998.
03.110)       .0001,VERIFY!
03.132)       "DERIVS"!  32!
03.133)       "AMPS"!  33!
03.135)       "FCN GENS"!  35!
03.998)   "PART 03 DONE"!
  "PART 06"!
06.001)       "IS ANALOG ON-LINE AND READY?"!  HALT!
06.010)       NORMAL!  @IX,@WA,X!   ICONS,USE!  1,C!
06.015)       @IC,M!
06.020)       CONN=0,  DIG=0
06.021)       "SET COEFFS"!  21!
06.025)       "SET FCN GENS?"! FG=0, FG=
06.026)       (FG==1)?  20!
06.100)       "ON-LINE VERIFY?"!  IV=0, IV=
06.101)       (IV==0)?  6.998.
06.102)       .0005,VERIFY!
06.499)
06.501)       "CONNECTION STMTS?"!  CONN=
06.505)       (CONN==0)?  6.601.
06.531)       "POTS"!  31!  HALT!
06.532)       "DERIVS"!  32!  HALT!
06.533)       "AMPS"!  33!  HALT!
06.535)       "FCN GENS"!  35!  HALT!
06.599)
06.601)       "CHECK AGAINST DIG?"!  DIG=
06.610)       (DIG==0)?  6.998.
06.621)       "COEFFS"!  21!  HALT!
06.622)       "DERIVS"!  22!  HALT!
06.623)       "AMPS"!  23!  HALT!
06.625)       "FCN GENS"!  25!  HALT!
06.998)   "PART 06 DONE"!
```

```
  "PART 10"!
10.001)     "CONSTANTS"
10.010)       L=6.0
10.020)       MOVL=0.55*10**(-3)
10.030)       MOVL=0.261*10**(-3)
10.040)       AOVL=0.8439*10**(-3)
10.050)       AB=0.84108*10**(-2)
10.060)       MB=0.814/*10**(-3)
10.070)       WB=0.1290
10.080)       N=4;N^
10.090)       LO=L/(N-1.0)
10.100)       W=MOVL*LO
10.110)       M3=MOVL*LO
10.120)       M2=M3
10.130)       ADMCO=-AOVL*LO
10.140)       W4=-(MOVL*LO/2.0+MB)
10.150)       M4= MOVL*LO/2.0 + MB
10.160)       ADM4CO=-(AOVL*LO/2.0 + AB)
10.165)     DRAGCO = -.00789*LO, DRAG4CO =-0.0102
10.170)       TIC=10
10.180)       TOP=200.
10.190)       THO= 10
10.200)       BETA=0.5
10.210)       GAMMA=(N-2.)*(TIC+TOP+THO)/TOP
10.212)     "SCALE FACTORS"
10.213)     AMAX = 45
10.214)     VMAX=3.5,    "VMAX<=3.5 TO SATISFY .015V**2<=F<=.0055A"
10.215)     FMAX = .15
10.220)     XMAX = 10
10.225)     XDDMAX = AMAX, YDDMAX = AMAX
10.230)     XDMAX = VMAX,  YDMAX = VMAX
10.235)     TMAX = 10*FMAX
10.240)     YMAX = XMAX
10.250)     LMAX=2.3,  DELXMAX=LMAX,  DELYMAX=LMAX
10.270)     DELLMAX = LMAX/10
10.299)
10.998)     "PART 10 DONE"!
  "PART 11"!
11.001)     "STATIC TEST VALUES"
11.002)     IC=0, SI=1
11.005)     "ADM=07"!  ADM=1, ADM=    "ADM=0 FOR EASIER ONLINE DEBUG"
11.006)     (ADM=-0)?  ADMCO = 0,  ADM4CO = 0 ,  "**********"
11.007)     K = 1.4,   M4 = .05
11.010)     X4D = 1, X3D = 1.5, X2D = 2.2
11.020)     Y4D = 1.1, Y3D = 1.7, Y2D = 1.6
11.040)     X4=4.75, X3=3.1,  X2=1.5
11.050)     Y4=3.9,  Y3=2.7,  Y2=1.4
11.998)   "PART 11 DONE"!
  "PART 12"!
12.001)     "INITIAL CONDITIONS"
12.005)     IC=1, SI=0
12.010)     X4D=0, X3D=0, X2D=0, Y4D=0, Y3D=0, Y2D=0
12.020)     X4=4.4674,  X3=2.9895,  X2=1.5004
12.040)     Y4=4.3617,  Y3=2.8955,  Y2=1.4415
12.041)       K=2.36
12.998)   "PART 12 DONE"!
  "PART 13"!
13.001)     "DIG EQUATIONS"
13.100)     "LENGTHS AND TENSIONS"
13.105)   X1=0,  Y1=0
13.111)     DELX1 = X2-X1,  DELY1 = Y2-Y1
13.112)     DELX2 = X3-X2,  DELY2 = Y3-Y2
13.113)     DELX3 = X4-X3,  DELY3 = Y4-Y3
13.121)     L1 = SQRT(DELX1**2 + DELY1**2)
13.122)     L2 = SQRT(DELX2**2 + DELY2**2)
13.123)     L3 = SQRT(DELX3**2 + DELY3**2)
13.131)     SIN1 = DELY1/L1,  COS1 = DELX1/L1
13.132)     SIN2 = DELY2/L2,  COS2 = DELX2/L2
13.133)     SIN3 = DELY3/L3,  COS3 = DELX3/L3
13.140)     DELL1 = L1-LO,  DELL2 = L2-LO,  DELL3 = L3-LO
```

```
13.161)     T1 = K*(DELL1/LO)*(1 + .5*DELL1/LO)
13.162)     T2 = K*(DELL2/LO)*(1 + .5*DELL2/LO)
13.163)     T3 = K*(DELL3/LO)*(1 + .5*DELL3/LO)
13.172)     T2X = COS2*T2 - COS1*T1,    T2Y = SIN2*T2 - SIN1*T1
13.173)     T3X = COS3*T3 - COS2*T2,    T3Y = SIN3*T3 - SIN2*T2
13.174)     T4X =         - COS3*T3,    T4Y =         - SIN3*T3
13.199)
13.300)     "DRAG TERMS"
13.310)     V2MNN = -X2D*SIN1 + Y2D*COS1
13.315)     V2PLN = -X2D*SIN2 + Y2D*COS2
13.320)     V3MNN = -X3D*SIN2 + Y3D*COS2
13.325)     V3PLN = -X3D*SIN3 + Y3D*COS3
13.330)     V4MNN = -X4D*SIN3 + Y4D*COS3
13.335)     D2MNN= DRAGCO*ABS(V2MNN)*V2MNN
13.340)     D2PLN= DRAGCO*ABS(V2PLN)*V2PLN
13.345)     D3MNN= DRAGCO*ABS(V3MNN)*V3MNN
13.350)     D3PLN= DRAGCO*ABS(V3PLN)*V3PLN
13.355)     D4MNN= DRAGCO*ABS(V4MNN)*V4MNN
13.360)     DBUOX = DRAG*CO*ABS(X4D)*X4D
13.365)     DBUOY = DRAG*CO*ABS(Y4D)*Y4D
13.370)     D2X = -SIN2*D2PLN - SIN1*D2MNN,   D2Y = COS2*D2PLN + COS1*D2
13.375)     D3X = -SIN3*D3PLN - SIN2*D3MNN,   D3Y = COS3*D3PLN + COS2*D3
13.380)     D4X =  DBUOX      - SIN3*D4MNN,   D4Y =   DBUOY    + COS3*D4
13.399)
13.400)     "SUM ALL NON ACCEL TERMS, IE DRAG+TENSION+WEIGHT"
13.402)     X2SUM = D2X+T2X,    Y2SUM = D2Y+T2Y - W
13.403)     X3SUM = D3X+T3X,    Y3SUM = D3Y+T3Y - W
13.404)     X4SUM = D4X+T4X,    Y4SUM = D4Y+T4Y + W4
13.499)
13.500)     "ADDED MASS TERMS FOR SOLVING SIM EQS"
13.502)     C2XX = ADMCO*(SIN2**2+SIN1**2)
13.503)     C3XX = ADMCO*(SIN3**2+SIN2**2)
13.504)     C4XX = ADM4CO*(      +SIN3**2)
13.512)     C2YY = ADMCO*(COS2**2+COS1**2)
13.513)     C3YY = ADMCO*(COS3**2+COS2**2)
13.514)     C4YY = ADM4CO*(      +COS3**2)
13.522)     C2XY = ADMCO*(-SIN2*COS2-SIN1*COS1)
13.523)     C3XY = ADMCO*(-SIN3*COS3-SIN2*COS2)
13.524)     C4XY = ADM4CO*(       -SIN3*COS3)
13.532)     M2X = M2-C2XX,    M2Y = M2-C2YY,    DEN2 = M2X*M2Y-C2XY**2
13.533)     M3X = M3-C3XX,    M3Y = M3-C3YY,    DEN3 = M3X*M3Y-C3XY**2
13.534)     M4X = M4-C4XX,    M4Y = M4-C4YY,    DEN4 = M4X*M4Y-C4XY**2
13.599)
13.600)     "SOL OF SIM EQS"
13.602)     X2DD = (C2XY*Y2SUM + M2Y*X2SUM) / DEN2
13.603)     X3DD = (C3XY*Y3SUM + M3Y*X3SUM) / DEN3
13.604)     X4DD = (C4XY*Y4SUM + M4Y*X4SUM) / DEN4
13.612)     Y2DD = (C2XY*X2SUM + M2X*Y2SUM) / DEN2
13.613)     Y3DD = (C3XY*X3SUM + M3X*Y3SUM) / DEN3
13.614)     Y4DD = (C4XY*X4SUM + M4X*Y4SUM) / DEN4
13.622)     F2X = M2*X2DD,     F2Y = M2*Y2DD
13.623)     F3X = M3*X3DD,     F3Y = M3*Y3DD
13.624)     F4X = M4*X4DD,     F4Y = M4*Y4DD
13.699)
13.700)     "NORMAL ACCEL FOR ADDED MASS"
13.710)     A2MN = -X2DD*SIN1 + Y2DD*COS1
13.720)     A2PL = -X2DD*SIN2 + Y2DD*COS2
13.730)     A3MN = -X3DD*SIN2 + Y3DD*COS2
13.740)     A3PL = -X3DD*SIN3 + Y3DD*COS3
13.750)     A4MN = -X4DD*SIN3 + Y4DD*COS3
13.759)
13.760)     "NORMAL AD MASS PLUS DRAG"
13.761)     F2MNN =  A2MN*ADMCO  + D2MNN
13.762)     F2PLN =  A2PL*ADMCO  + D2PLN
13.763)     F3MNN =  A3MN*ADMCO  + D3MNN
13.764)     F3PLN =  A3PL*ADMCO  + D3PLN
13.765)     F4MNN =  A4MN*ADM4CO + D4MNN
13.799)
13.800)     "ADDITIONAL ANALOG VBLS"
```

```
13.810)      STRCO = K*DELLMAX/(LO*TMAX)
13.811)      STR1 = STRCO*(1 + .5*DELL1/LO)
13.812)      STR2 = STRCO*(1 + .5*DELL2/LO)
13.813)      STR3 = STRCO*(1 + .5*DELL3/LO)
13.820)      KFEEDBK = DELXMAX/XMAX = 1/10          , "ASSUMES KFDBKX=KFDBKY
13.830)      STRPOT = .5*K/LO**2 * (DELLMAX**2/TMAX)
13.840)      ADMPOT = -ADMCO*AMAX/FMAX
13.845)      ADM4POT = -ADM4CO*AMAX/FMAX /10
13.899)
13.900)      "FCN GEN SETTINGS"
13.901)      FGSET = -DRAGCO*VMAX**2/FMAX
13.902)      FG4SET = -DRAG4CO*VMAX**2/FMAX
13.903)      RAT4 = FG4SET/FGSET
13.904)      AC,ADDR,NBPT,X4I6S,FX4I6S,ERR,KN;
13.905)      I=1,1,16!   X4I5 = .125*(I-8)
13.910)      I = 1,1,16!   FGDRG4I5 =    FGSET*X4I5*ABS(X4I5)
13.915)      I = 1,1,16!   FGDRG44I5 = RAT4*FGDRG4I5
13.920)      FGDR34I6S = .9999
13.998)   "PART 13 DONE"!
   "PART 20"!
20.001)      "SET FCN GENERATORS"
20.010)      "*****KUMMON STMNT IN 13 TO GET IT BEFORE DEF OF X4I5"
20.020)      AC=ICONS, NBPS=16
20.040)      I = 1,1,16!   FX4I5 =    FGDRG4I5
20.100)      FG100=1, ADDR=FG100, ERR=0, ADDR!   @SDFG,L;
20.101)      (ERR<0)?   "CONS SELECT ERROR,", ERR!
20.102)      (ERR>0)?   "DCFG SETUP ERROR,", ERR!
20.200)      FG101=2, ADDR=FG101, ERR=0, ADDR!   @SDFG,L;
20.201)      (ERR<0)?   "CONS SELECT ERROR,", ERR!
20.202)      (ERR>0)?   "DCFG SETUP ERROR,", ERR!
20.300)      FG110=3, ADDR=FG110, ERR=0, ADDR!   @SDFG,L;
20.301)      (ERR<0)?   "CONS SELECT ERROR,", ERR!
20.302)      (ERR>0)?   "DCFG SETUP ERROR,", ERR!
20.400)      FG111=4, ADDR=FG111, ERR=0, ADDR!   @SDFG,L;
20.401)      (ERR<0)?   "CONS SELECT ERROR,", ERR!
20.402)      (ERR>0)?   "DCFG SETUP ERROR,", ERR!
20.500)      FG120=5, ADDR=FG120, ERR=0, ADDR!   @SDFG,L;
20.501)      (ERR<0)?   "CONS SELECT ERROR,", ERR!
20.502)      (ERR>0)?   "DCFG SETUP ERROR,", ERR!
20.599)
20.600)      I = 1,1,16!   FX4I5 =    FGDRG44I5
20.700)      FG130=7, ADDR=FG130, ERR=0, ADDR!   @SDFG,L;
20.701)      (ERR<0)?   "CONS SELECT ERROR,", ERR!
20.702)      (ERR>0)?   "DCFG SETUP ERROR,", ERR!
20.800)      FG131=8, ADDR=FG131, ERR=0, ADDR!   @SDFG,L;
20.801)      (ERR<0)?   "CONS SELECT ERROR,", ERR!
20.802)      (ERR>0)?   "DCFG SETUP ERROR,", ERR!
20.998)   "PART 20 DONE"!
   "PART 21"!
21.001)        1E001=XDDMAX/XDMAX/10*BETA
21.002)      1E002 = X4D/XDMAX
21.003)      1E003 = XDMAX/XMAX * BETA
21.010)      1E010 = X4/XMAX
21.011)      1E011 = FMAX/(M4*XDDMAX) /10  *2
21.012)      1E012 = -M4/(M4*YDDMAX)   /10
21.013)      1E013 = YDDMAX/YDMAX / 10  * BETA
21.020)      1E020 = Y4D/YDMAX
21.021)      1E021 = YDMAX/YMAX  * BETA
21.022)      1E022 = Y4/YMAX
21.023)      1E023 = KFEEDBK
21.030)      1E030 = FMAX/(M3*XDDMAX) /10
21.031)        1E031=XDDMAX/XDMAX/10*BETA*GAMMA/10
21.033)        1E033=XDMAX/XMAX*BETA*GAMMA/100
21.041)      1E041 = FMAX/(M3*XDDMAX) /10
21.042)        1E042=YDDMAX/YDMAX/10*BETA*GAMMA /10
21.050)        1E050=YDMAX/YMAX/10*BETA*GAMMA /10
21.110)        1E110=KFEEDBK
21.122)        1E122= -X3/XMAX
21.123)      1E123 = LO/LMAX
21.130)      1E130 = STRPOT
```

```
21.131)      1E131 = K*DELLMAX/(LD*TMAX)
21.132)      1E132 = STRPOT
21.140)      1E140 = W/FMAX
21.141)      1E141 = ADM*POT
21.142)      1E142 = ADMPOT
21.143)      1E143 = ADMPOT
21.150)        1E150 = -X2/XMAX
21.200)        1E200=KFEEDBK
21.201)        1E201=KFEEDBK
21.313)        1E313=Y3D/YDMAX
21.320)        1E320=X3D/XDMAX
21.340)        1E340=Y3/YMAX *(-1)
21.343)        1E343=Y2/YMAX *(-1)
21.900)      1E000 = FMAX/(M4*XDDMAX)   /10
21.998)   "PART 21 DUNE"!
  "PART 22"!
22.022)      1D022 = -X3DD/XDMAX / 10 *GAMMA*BETA
22.030)      1D030 = X3D/XMAX / 10   *GAMMA*BETA
22.032)      1D032 = -Y3DD/YDMAX / 10 *GAMMA*BETA
22.040)      1D040 = Y3D/YMAX / 10   *GAMMA*BETA
22.402)        1D402=-X4DD/XDMAX/10*BETA
22.410)        1D410= X4D/XMAX/10*BETA
22.412)        1D412= -Y4DD/YDMAX/10 * BETA
22.420)        1D420= Y4D/YMAX/10 * BETA
22.899)
22.900)      "DUMMY SECTION TO MOVE PART 23 PAGE BOUNDARY"
22.901)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.902)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.903)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.904)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.904)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.905)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.906)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.907)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.908)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.909)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.910)      "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
22.998)   "PART 22 DUNE"!
  "PART 23"!
23.001)      1A001 = -Y4DD/YDDMAX
23.003)      1A003 = DELY3/DELYMAX
23.011)      1A011 = DELX3/DELXMAX
23.013)      1A013 = -Y3DD/YDDMAX
23.021)        1A021=0.0
23.022)      1A022 = X3D/XDMAX
23.023)        1A023=0.0
23.030)      1A030 = -X3/XMAX
23.031)      1A031 = DELY2/DELYMAX
23.032)      1A032 = Y3D/YDMAX
23.040)      1A040 = -Y3/YMAX
23.041)        1A041=X3/XMAX
23.042)        1A042=DELX2/DELXMAX
23.043)        1A043=X2/XMAX
23.051)      1A051 = -DELL3/DELLMAX
23.052)        1A052=SIN3
23.100)      1A100 = -X3DD/XDDMAX
23.103)      1A103 = T3/TMAX
23.104)      1A104 = T2/TMAX
23.113)      1A113 = SIN3*X4D/XDMAX
23.114)      1A114 = SIN3*X3D/XDMAX
23.115)      1A115 = SIN2*X3D/XDMAX
23.120)        1A120=-X3D/XDMAX
23.123)        1A123=-SIN3
23.124)        1A124=-COS3
23.125)      1A125 = -COS3*Y4D/YDMAX
23.130)        1A130=-Y3D/YDMAX
23.131)        1A131=-(L2*L2)/(LMAX*LMAX)
23.133)      1A133 = -COS3*Y3D/YDMAX
23.134)      1A134 = -COS2*Y3D/YDMAX
```

```
23.135)          1A135=  (DELX2*DELX2)/(DELXMAX*DELXMAX)
23.140)          1A140=Y3/YMAX
23.141)          1A141=X3/XMAX
23.144)          1A144=L2/LMAX
23.145)       1A145 = SIN3*X3DD/XDDMAX
23.153)       1A153 = SIN2*X3DD/XDDMAX
23.154)          1A154=-SIN2
23.155)          1A155=-COS2
23.200)          1A200=-DELL2/DELLMAX
23.201)          1A201=0.0
23.202)       1A202 = STR2
23.203)          1A203=0.0
23.210)          1A210=Y3D/YDMAX
23.211)       1A211 = F3X/FMAX
23.212)       1A212 = F3Y/FMAX
23.220)          1A220=X3D/XDMAX
23.221)       1A221 = F4X/FMAX
23.222)       1A222 = (F4Y-W4)/FMAX /2
23.223)       1A223 = A4MN/AMAX
23.230)       1A230 = F4MNN/FMAX
23.231)       1A231 = A3PL/AMAX
23.232)       1A232 = F3PLN/FMAX
23.233)       1A233 = A3MN/AMAX
23.240)       1A240 = F3MNN/FMAX
23.241)          1A241=Y3/YMAX
23.242)          1A242=-Y3/YMAX
23.243)          1A243=Y2/YMAX
23.250)          1A250=-X3/XMAX
23.251)       1A251 = V4MNN/VMAX
23.252)       1A252 = V3PLN/VMAX
23.253)       1A253 = V3MNN/VMAX
23.303)       1A303 = -COS3*Y4DD/YDDMAX
23.304)       1A304 = -COS3*Y3DD/YDDMAX
23.305)       1A305 = -COS2*Y3DD/YDDMAX
23.310)          1A310=-(L3+L3I/(LMAX+LMAX)
23.313)          1A313=( DELX3*DELX3I/(DELXMAX*DELXMAX)
23.314)          1A314=L3/LMAX
23.315)       1A315 = SIN3*F3PLN/FMAX
23.323)       1A323 = SIN2*F3MNN/FMAX
23.324)       1A324 = -COS3*T3/TMAX
23.325)       1A325 =  COS2*T2/TMAX
23.341)          1A341=-Y3/YMAX
23.342)          1A342=-X3/XMAX
23.343)       1A343 = SIN3*F4MNN/FMAX
23.344)       1A344 = -COS3*F3PLN/FMAX
23.401)          1A401= SIN3
23.402)          1A402=X4D/XDMAX
23.403)          1A403=-SIN3
23.410)          1A410=-X4/XMAX
23.412)          1A412=Y4D/YDMAX
23.413)          1A413=-COS3
23.420)          1A420=-Y4/YMAX
23.423)          1A423=COS3*T3/TMAX
23.433)          1A433=SIN3*T3/TMAX
23.534)          1A534=-COS3*F4MNN/FMAX
23.541)       1A541 = -COS2*F3MNN/FMAX
23.542)       1A542 = -SIN3*T3/TMAX
23.544)       1A544 = SIN2*T2/TMAX
23.554)          1A554=-SIN3*X4DD/XDDMAX
23.601)       1A601 = Y3/YMAX
23.603)       1A603 = +COS3
23.604)       1A604 = +SIN2
23.614)       1A614 = X3/XMAX
23.631)          1A631=SIN3
23.632)          1A632=COS3
23.700)       1A700 = +COS2
23.900)       1A000 = -X4DD/XDDMAX
23.998)    "PART 23 DONE"!
    "PART 25"!
25.100)       1F100 =-J*MNN/FMAX
```

```
25.101)      1F101  =-D3PLN/FMAX
25.110)      1F110  =-D2PLN/FMAX
25.111)      1F111  =-D2MNN/FMAX
25.120)      1F120  =-D3MNN/FMAX
25.130)      1F130  =  -DBUOX/FMAX
25.131)      1F131  =  -DBUOY/FMAX
25.998)  "PART 25 DONE"!
  "PART 31"!
31.001)      1P001  =  1E001*1A000
31.002)      1P002  =  1E002*(-1.)
31.003)      1P003  =  1E003*1A402
31.010)      1P010  =  1E010*1.0
31.011)      1P011  =  1E011*1A222
31.012)      1P012  =  1E012*(-1.0)
31.013)      1P013  =  1E013*1A001
31.020)      1P020  =  1E020*(-1.0)
31.021)      1P021  =  1E021*1A412
31.022)      1P022  =  1E022*1.0
31.023)      1P023  =  1E023*1A011
31.030)      1P030  =  1E030*1A211
31.031)      1P031  =  1E031*1A100
31.033)      1P033  =  1E033*1A022
31.041)      1P041  =  1E041*1A212
31.042)      1P042  =  1E042*1A013
31.050)      1P050  =  1E050*1A032
31.110)      1P110  =  1E110*1A031
31.122)      1P122  =  1E122
31.123)      1P123  =  1E123*(-1.0)
31.130)      1P130  =  1E130*1A081
31.131)      1P131  =  1E131*(-1.0)
31.132)      1P132  =  1E132*1A200
31.140)      1P140  =  1E140*1.0
31.141)      1P141  =  1E141*1A223
31.142)      1P142  =  1E142*1A231
31.143)      1P143  =  1E143*1A233
31.150)      1P150  =  1E150
31.200)      1P200  =  1E200*1A002
31.201)      1P201  =  1E201*1A042
31.313)        1P313=-1E313
31.320)        1P320=-1E320
31.340)        1P340=  1E340
31.343)        1P343=  1E343
31.900)      1P000  =  1E000*1A221
31.998)  "PART 31 DONE"!
  "PART 32"!
32.022)      1D022  =    10*1P031 / 10
32.030)        1D030=10*1P033/10
32.032)      1D032  =    10*1P042 / 10
32.040)        1D040=10*1P050/10
32.402)        1D402=10*1P001/10
32.410)        1D410=1P003/10
32.412)        1D412=10*1P013/10
32.420)        1D420=1P021/10
32.998)  "PART 32 DONE"!
  "PART 33"!
33.001)      1A001  =  -(10*1P012*10*1P011)
33.003)      1A003  =  -(10*1A420 *10*1A601 *1P200*10)
33.011)      1A011  =  -(10*1A410 *10*1A614 *10*1P023)
33.013)      1A013=-1P041*10
33.021)      1A021  =0
33.022)      1A022  =  -1A120
33.023)      1A023=0
33.030)      1A030=-1A141
33.031)      1A031=-(10*1A040*10*1P110*10*1A243)
33.032)      1A032=-1A139
33.040)      1A040=-1A140
33.041)      1A041=-1P122
33.042)      1A042=-(10*1A030*1A043*10*10*1P201)
33.043)      1A043=-1P150
```

```
33.051)    1A051=-(10*1A314+10*1P123)
33.052)    1A052=-(1P130 + 1P131)
33.100)    1A100 = -10*1P030
33.103)    1A103=-1A051*1A052
33.104)    1A104=-1A200*1A202
33.113)    1A113=-1A403*1A402
33.114)    1A114=-1A123*1A022
33.115)    1A115=-1A022*1A154
33.120)    1A120=-1A220
33.123)    1A123=-1A003/1A314
33.124)    1A124=-1A011/1A314
33.125)      1A125=-1A412*1A632
33.130)    1A130=-1A210
33.133)    1A133=-1A603*1A032
33.134)    1A134=-1A700*1A032
33.135)    1A135= 1A042*1A042
33.140)    1A140=-1A242
33.141)    1A141=-1A250
33.144)    1A144=SQR(-1A131)
33.145)    1A145=-1A401*1A100
33.153)    1A153=-1A100*1A604
33.154)    1A154=-1A031/1A144
33.200)    1A200=-(10*1A144+10*1P123)
33.201)    1A201=0
33.202)    1A202=-(1P132 + 1P131)
33.203)    1A203 = 0
33.210)    1A210=-(1A201+1P313)
33.211)    1A211=-(1A315+1A323+10*1A325+10*1A325)
33.212)    1A212=-(1A344+1A541+10*1A542+10*1A544+1P140)
33.220)    1A220=-(1A021+1P320)
33.221)    1A221=-(10*1A423+1A343+1P129)
33.222)    1A222=-(1A534+10*1A433 +1P133) + (.5)
33.223)    1A223=-1A554-1A303
33.230)    1A230=-(10*1P141 +1P100)
33.231)    1A231=-1A145-1A304
33.232)    1A232=-(1F101+1P142)
33.233)    1A233=-(1A153+1A305)
33.240)    1A240=-(1P143+1F120)
33.241)    1A241=-1P340
33.242)    1A242=-1A241
33.243)    1A243=-1P343
33.250)    1A250=-1A041
33.251)    1A251=-1A113 -1A125
33.252)    1A252=-1A114-1A133
33.253)    1A253=-1A115-1A134
33.303)    1A303=-1A413*1A001
33.304)    1A304=-1A124*1A013
33.305)    1A305=-1A013*1A155
33.313)    1A313= 1A011*1A011
33.314)    1A314=SQR(-1A310)
33.315)    1A315=-1A123*1A232
33.323)    1A323=-1A240*1A154
33.324)    1A324=-1A603*1A103
33.325)    1A325=-1A155*1A104
33.341)    1A341=-1A241
33.342)    1A342=-1A041
33.343)    1A343=-1A403*1A230
33.344)    1A344=-1A603*1A232
33.401)    1A401=-1A123
33.402)    1A402 = -1P002
33.403)    1A403=-1A401
33.410)    1A410 = -1P010
33.412)    1A412 = -1P020
33.420)    1A420 = -1P022
33.423)    1A423=-1A324
33.433)    1A433=-1A542
33.534)    1A534=-1A632*1A230
33.541)    1A541=-1A240*1A700
33.542)    1A542=-1A401*1A103
33.544)    1A544=-1A154*1A104
```

```
33.554)         1A554= -1A631*1A000
33.603)         1A603=-1A124
33.604)         1A604=-1A154
33.614)         1A614 = -1A030
33.631)         1A631=-1A402
33.632)         1A632=-1A413
33.700)         1A700=-1A185
33.900)         1A000 = -(10*1P000)
33.998)   "PART 33 DONE"!
   "PART 35"!
35.100)       1F100 = ABS(1A251)#1A251*FGSET
35.101)       1F101 = ABS(1A252)*1A252*FGSET
35.120)       1F120 = ABS(1A253)*1A253*FGSET
35.130)         1F130= ABS(1A402)*1A402*FGSET
35.131)         1F131=ABS(1A412)*1A412*FGSE!
35.998)   "PART 35 DONE"!
   "PART 80"!
80.001)       "PRINT OPTIONS"  NORMAL!
80.010)       "PRINT VAR LIST?"!  PR=0, PR=
80.020)       (PR==0)?  80.998:
80.030)       "VAR LIST ON LP1"!
80.040)       @LP1,0!  @D,F!
80.050)       "MOI VARIABLE LIST"!  1.001! 1.002!  @VAR!
80.060)       @!1,0!
80.998)   "PART 80 DONE"!
   "PART 90" !
90.001)       "USEFUL COMMANDS"
90.010)       "LP1 ASSIGNS"!  @S,@LP1,0!  @D,@LP1,0!
90.020)       "T1 ASSIGNS"!  @S,@T1,0!  @D,@T1,0!
90.040)       "PROGRAM LISTING ON LP1"!  @LP1,0!  @D,F!  @PAR!  @T1,0!
"MOI INPUT DONE"!
DONE!
```

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.